CONTEXT AWARE PRE-CRASH SYSTEM FOR VEHICULAR AD HOC NETWORKS USING DYNAMIC BAYESIAN MODEL

PhD Thesis

Musaab Zeyad A. R. Aswad

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Software Technology Research Laboratory

De Montfort University

Leicester - United Kingdom

March - 2014

Dedication

To my beloved father

Prof. Zeyad A.R. Aswad

Without him, my PhD dream would not be a reality.

This thesis is a reminder that all my life I will be thanking heaven for a special father like you.

To my loving mother

Asmaa M. Al-Mashhadani

For her special love, ceaseless prayers and encouragement.

Thank you for everything you have done for me since the day I was born.

To my loving wife

Hiba F. Kubba

For her special patience, continual prayers and the love she gives me.

Thank you for everything you have done for me since the day I met you.

Abstract

Tragically, traffic accidents involving drivers, motorcyclists and pedestrians result in thousands of fatalities worldwide each year. For this reason, making improvements to road safety and saving people's lives is an international priority. In recent years, this aim has been supported by Intelligent Transport Systems, offering safety systems and providing an intelligent driving environment. The development of wireless communications and mobile ad hoc networks has led to improvements in intelligent transportation systems heightening these systems' safety. Vehicular ad hoc Networks comprise an important technology; included within intelligent transportation systems, they use dedicated short-range communications to assist vehicles to communicate with one another, or with those roadside units in range. This form of communication can reduce road accidents and provide a safer driving environment.

A major challenge has been to design an ideal system to filter relevant contextual information from the surrounding environment, taking into consideration the contributory factors necessary to predict the likelihood of a crash with different levels of severity. Designing an accurate and effective pre-crash system to avoid front and back crashes or mitigate their severity is the most important goal of intelligent transportation systems, as it can save people's lives. Furthermore, in order to improve crash prediction, context-aware systems can be used to collect and analyse contextual information regarding contributory factors.

The crash likelihood in this study is considered to operate within an uncertain context, and is defined according to the dynamic interaction between the driver, the vehicle and the environment, meaning it is affected by contributory factors and develops over time. As a crash likelihood is considered to be an uncertain context and develops over time, any usable technology must overcome this uncertainty in order to accurately predict crashes.

This thesis presents a context-aware pre-crash collision prediction system, which captures information from the surrounding environment, the driver and other vehicles on the road. It utilises a Dynamic Bayesian Network as a reasoning model to predict crash likelihood and severity level, whether any crash will be fatal, serious, or slight. This is achieved by combining the above mentioned information and performing probabilistic reasoning over time.

The thesis introduces novel context aware on-board unit architecture for crash prediction. The architecture is divided into three phases: the physical, the thinking and the application phase; these which represent the three main subsystems of a context-aware system: sensing, reasoning and acting. In the thinking phase, a novel Dynamic Bayesian Network framework is introduced to predict crash likelihood. The framework is able to perform probabilistic reasoning to predict uncertainty, in order to accurately predict a crash. It divides crash severity levels according to the UK department for transport, into fatal, serious and slight.

GeNIe version 2.0 software was used to implement and verify the Dynamic Bayesian Network model. This model has been verified using both syntactical and real data provided by the UK department for transport in order to demonstrate the prediction accuracy of the proposed model and to demonstrate the importance of including a large amount of contextual information in the prediction process.

The evaluation of the proposed system delivered high-fidelity results, when predicting crashes and their severity. This was judged by inputting different sensor readings and performing several experiments. The findings of this study has helped to predict the probability of a crash at different severity levels, accounting for factors that may be involved in causing a crash, thereby representing a valuable step towards creating a safer traffic network.

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the Software Technology Research Laboratory (STRL) at De Montfort University, United Kingdom.

No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

This thesis was written by myself and produced using MS Word 2013.

Musaab Zeyad A.R. Aswad

Acknowledgments

In the Name of Allah, Most Gracious, Most Merciful

All praises to Allah for His strength and His blessing, without which, throughout my entire life, I would not even be in this place and have never completed this thesis.

My appreciation goes to Prof. Hussein Zedan, the head of the STRL, for his guidance throughout my PhD study. This thesis would not have been completed without the help, support, guidance and efforts of my supervisor, Dr. Ali Hilal Al-Bayatti, so I would also like to thank him for his professional guidance, perceptive suggestions regarding the research work and constant encouragement. He was the first to help and support me when I started pursuing my PhD.

Special thanks are also due to my mentor, Prof. Hilal Al-Bayatti, Dr. Waheeb Al-Khaja, Dr. Jamal Al-Sultan, Dr. Nihad Al-Juborri and Dr. Saif AL-Sultan for their attention, cooperation, support, comments and constructive criticism.

Very special thanks to my parents, whom I owe for everything I am today. My father, Prof. Zeyad A.R. Aswad, instilled in me the love of reading and the pursuit of knowledge from my early sapling years; to him I dedicate this thesis. If it were not for him, I would not be sitting here today. Moreover, I would like to thank my mother Asmaa; without her endless support, prayers and guidance, I would not be the man I am today. Thank you Mom for everything you have done for me since the day I was born.

Finally, I would like to thank my sweet lovely wife Hiba for her continual prayers, encouragement and the love she gives me. I would also like to thank my kids; my heroes Omar, Abdulla and my little angle Lina, whose love is worth it all.

Thank you all.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

AI	Artificial Intelligence
API	Application Program Interface
ASR	Automatic Speech Recognition
AU	Application Unit
BBN	Bayesian Belief Network
BNT	Bayes Net Toolbox
CALM	Continuous Air interface for Long and Medium range
CAS	Context-aware Systems
CASS	Context-awareness Sub-Structure
CCD	Charged Coupled Device
CCH	Control Channel
CM	Context Manager
CoBrA	Context Broker Architecture
CPT	Conditional Probability Table
CR	Context Reasoner
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DfT	Department for Transport
DSRC	Dedicated Short Range Communication
EEBL	Emergency Electronic Break lights
GMTk	Graphical Model Toolkit
GPRS	General Packet Radio Service
GPS	Global Position System
GSM	Global System for Mobile communications
HMI	Human Machine Interface
HMM	Hidden Markov Models
HSDPA	High-Speed Downlink Packet Access
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transportation Systems
IU	Interface Unit
MANET	Mobile Ad Hoc Network
OBU	On Board Unit
PDA	Personal Digital Assistant
RCP	Resource Command Processor
RSU	Road Side Unit
TMC	Traffic Management Centre
UML	Untied Modelling Language

V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular ad hoc Network
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Chapter 1 - Introduction

Chapter Objectives:

1. Give an introduction to and explain the motivation of this research.
2. List the research questions.
3. Present the research methodology.
4. Outline the thesis structure.

1.1 Motivation

In the UK in 2012, the department for transport (DfT) reported that 1,900 people had been killed and 24,430 seriously injured due to crashes; this statistic was a mere proportion of the 204,350 casualties (slight injuries, serious injuries and fatalities). The total reported number of child casualties (those aged 0-15) killed or seriously injured was (KSI) 19,890. Road injury accidents reported to the police numbered approximately 151,250, while the fatal accident figure was 1,780 [1, 2]. All this statistical information actually represents only a cautious improvement on reports issued for 2010 and 2011, despite the use and activation of Intelligent Transportation Systems (ITS). Therefore, there is evidently still an abundant need to improve road safety and ITS efficiency further, in order to prevent road crashes, reduce the number of fatalities and save people’s lives. Accordingly, governments, vehicle manufacturers and researchers are seeking to find and develop new technologies in order to achieve this goal. All the applications available depend for their success on the accuracy of information collected from the surrounding environment, other vehicles, live traffic and roadside devices. Indeed, yet no comprehensive system exists that is able to reliably monitor all the potentially dangerous situations that may arise on the road; i.e. real contributory factors (such as road environment, vehicle defects, behaviour or inexperience and driver error or reactions). Pre-crash systems constitute a key technology that is anticipated to have far-reaching implications for the future of mass transportation.

The available data cannot work effectively without a reliable communication method to connect the nodes transmitting information to all those in range. For this purpose Vehicular ad hoc Networks (VANET), which use Dedicated Short Range Communication DSRC/WAVE technology, offer enhanced and controlled communication, able to support vehicle to vehicle (V2V) communication and vehicle to infrastructure (V2I) communication. These forms of communication integrate a wide range of VANET applications; of these, safety applications are the most important for enhancing road safety to improve and save people’s lives by reducing the number of injuries resulting from road crashes [3].

1.2 Thesis scope

This thesis develops a system that can work in all environment related conditions (road surface, traffic density status, and weather condition etc.) and on any type of road (dual carriageway, one way street, one way street slip road, roundabout, single carriageway and slip road) according to the DfT categorization. In addition, the proposed system take into it account the driver age and gender. Moreover, the most important vehicle factors that have the heavy weight in the causes of any crash.

The proposed system intend to have the capability to predict the front and rear crashes and their level of severity, by reasoning the sensed contextual information using Dynamic Bayesian Network (DBN) to predict the crash severity either to be fatal, serious or slight. The proposed system should deploy a prevention action that must deploy after predicting the crash, but this part of the system is consider to be the future work and it is not covered in this thesis. This is because the complexity of the proposed system and it is need another algorithm to be responsible of deploying the suitable solution or action to avoid and prevent the predicted crash.

The proposed system is suitable to apply in any country or region, and the only thing that should done, is update the values of the DBN conditional probability table of network nodes with the new values from the destination country.

1.3 Research Questions

In the literature review the current available predication systems has been analysed and determined the main research question as follows:

"How to predict a vehicle crash before it happens in VANET utilising the context-aware system approach?"

The aim of the work is to address the question above efficiently. To achieve this it is logical to partition the research question into sub-questions, as this makes it easier to tackle each one individually. These questions can be summarised as follows:

- ***How to design an effective Pre-crash system architecture for VANET by utilising a context-aware system approach?***
- ***How to design an efficient Pre-crash system that can perform reasoning over time and under uncertainty?***
- ***What kind of information is needed to predict crash severity accurately?***

1.4 Research Methodology

To achieve the desired objectives for the proposed system, certain methodologies based on Kendall and Kendall's system development life cycle concept (SDLC) should be followed [4], as these offer a systematic approach to the design and implementation of any system:

1. Identify the research problems and recognise it, in accordance with the opportunities and objectives accomplished by the literature survey.

2. Determining information requirements and data preparation methods to solve these problems.
3. Analysing system needs and searching for the best technique to resolve research problems.
4. Designing and applying the selected technique, to improve the system.
5. Documentation, testing and maintaining the model.
6. Implementing and evaluating the system so as to specify the strengths and the weaknesses of the concept.
7. Determining the possible directions of future work.

The investigation in this thesis is based on a three stage literature review, covering: VANET, Context Aware Systems (CAS) and also offering an overview of current Pre-crash systems. First of all VANET identification, architecture, communications and applications are presented. Then an overview of context and context-aware systems is presented too. Finally, the existing Pre-crash systems are examined in depth, as are their main weaknesses and lack thereof, as our aim in this research is to design a novel context-aware pre-crash system in VANET to predict crashes early and aid in preventing and avoiding accidents to save people's lives, as shown in chapter 2.

1.5 Measurable Outcomes

Success in regards to the work reported in this research will be judged as follows:

- The research questions established at the beginning of this thesis have to be met.
- A study presenting how the proposed architecture can be applied in VANET, in order to predict vehicle crashes has to be conducted.
- A study showing how our proposed system differs from others will be illustrated.

- Adding more scenarios and number of nodes to the contributory factors which affect the design.

1.6 Contributions

In this thesis, the following contributions were present:

(i) Pre-crash system built upon a DBN model that operates under various conditions and depending in three main contributory factors:

- Vehicle: e.g. speed, distance, brake pedal and lane.
- Driver: gender and age.
- Environment: e.g. road surface, traffic density, road type, timing, day, and road lightening.

(ii) Pre-crash system architecture built upon a five-layer conceptual framework [5, 6]. Intended to predict the crash by sensing the contributory factors in the sensing layer, then reasoning these inputs and then to apply the suitable application in the application layer trying to avoid or mitigate the crash.

(iii) The proposed system is also intend to have the capacity to make decisions before a collision occurs by processing contextual information and applying reasoning using a DBN.

1.7 Thesis structure

According to the objective of the research, this section provides a summary of the remaining chapters in this thesis based on each chapter's contents:

Chapter 2: Overview of VANET and Context-Aware systems

This chapter provides an overview of the VANET: architecture, communication, applications and services. The Context-Aware System (CAS) is also explained in terms of: context, definition, sensing, modelling and uncertain reasoning using different algorithms. In addition, there is a justification for the reasoning algorithm that is used in this thesis.

Finally, an investigation is presented about existing pre-crash systems that have been executed in the field of crash prediction and prevention to explain their lack, and how the proposed system will be different from the carried out systems.

Chapter 3: Preliminaries

This chapter is divided into three parts. Firstly, it describes the main aspects of the pre-crash system and its aims. In addition, it categorises the elements of the contributory factors and defines the stages pertaining to each crash level, and how the pre-crash system predicts these crash levels from the perspective of a context-aware system. It also illustrates the assumptions made when designing a pre-crash system. Secondly, an overview of the DBN method is given with an explanation of its ability to predict a crash as the reasoning method, which has been used in this thesis. Finally, an overview of some existing software packages is given and used to implement any DBN, with a justification of why the GeNIe software has been chosen to implement the proposed system.

Chapter 4: Context Aware On Board Unit Architecture

This chapter a presents the On Board Unit (OBU) architecture for the proposed pre-crash system and details its crash prediction mechanism in VANET. The OBU is designed based on certain aspects of the context-aware system as affected by three contributory factors; all system components are described with an explanation of how they interact to predict the crash.

Chapter 5: Pre-crash System Design and Development

This chapter presents in detail all the steps necessary to design the DBN as reasoning method to predict a crash. The first steps, are choosing the DBN nodes, then the DBN graph and CPTs design, and finally the inference step for the hypothesis node.

Chapter 6: Pre-crash System Validation and Evaluation

This chapter demonstrates the proposed system validity using real and synthetic data. It clarifies how the pre-crash system predicts the crash and the level of severity concerning all possible cases. Furthermore, it presents the evaluation by presenting experiments to show the system's ability to predict crashes severity level.

Chapter 7: Conclusion and Future works:

This chapter presents a summary of the work in this thesis, and summarises the results. Then it draws some conclusions and gives suggestions for future work that proceeds from this work in light of the limitations of our approach.

Chapter 2 - Overview of VANET and Context-Aware Systems

Chapter Objectives:

1. Provide an overview of Vehicular ad hoc Network (VANET).
2. Present the concept of Context Aware System (CAS).
3. Investigate existing Pre-crash systems.

2.1 Ad-Hoc Networks

Ad-hoc networks have emerged as one of the most attractive research for the network community over the last few years. Ad-hoc network consists of a set of nodes that are equipped with a wireless interface, and which are able to communicate with each other in the absence of any kind of network infrastructure. One of the most important features of ad-hoc networks is their capacity for wireless multi-hop communication. This feature led to the evolution of a number of specialised arrangements for different kinds of networks, such as Mobile Ad Hoc Network (MANET), Wireless Mesh Network (WMN), Wireless Sensor Network (WSN) and Vehicle ad-hoc Network (VANET) [3, 7, 8].

Mobile Ad-Hoc Networks (MANETs) are wireless networks allowing mobile computing devices to communicate without any support from a fixed infrastructure. The nodes in MANET are self-organised and arbitrary; these nodes are usually battery-operated [8]. Two nodes can communicate directly with each other if they are within radio range and can communicate using multi-hop routing if they are within different ranges [9, 10]. As new wireless devices come into existence this technology becomes more extensive and successful on the futures markets.

2.2 Vehicular Ad Hoc Networks

Vehicular Ad Hoc Networks (VANETs) are a special type of Mobile Ad Hoc Networks (MANETs). These are organised according to how their nodes communicate without the need for any infrastructure; each node in the network can be either a source or a destination, but nodes sometimes works better as routers [11, 12].

New standards involving dedicated short-range communications (DSRC), a form of wireless communication medium comprising short-range communication technology, are emerging to regulate VANET communication between moving vehicles on the road. DSRC can provide two types of communication: Vehicle to Vehicle (V2V) or Vehicle to Infrastructure (V2I) [13]. The infrastructure is represented with an RSU and communicates with vehicles using on board units (OBUs) [14].

2.2.1 VANET Architecture

VANET System architecture is a high-level abstraction with three major components: OBU, RSU and an Application Unit (AU):

a) Application Unit (AU):

The AU is a dedicated device that insures each vehicle in the network must be equipped to use the services provided by other nodes (vehicles or RSUs). The AU executes a single or a set of applications, such as safety applications and general services (i.e. Internet). It is also used to store information for each vehicle [15-17]. The AU uses the OBU's communication capabilities and can be an integrated part of a vehicle or a portable device such as a laptop or PDA.

b) Road Side Unit (RSU):

RSUs are devices provided at dedicated locations along the road (for example at intersections, on motorways, at traffic lights, in areas of congestion or in car parks), as shown in Figure 2.1. They are equipped with dedicated short range communication devices based on IEEE 802.11p technology that communicates with any OBUs in range. RSUs are typically equipped to link to other hosts or the Internet to increase the network's range of communication [18, 19].

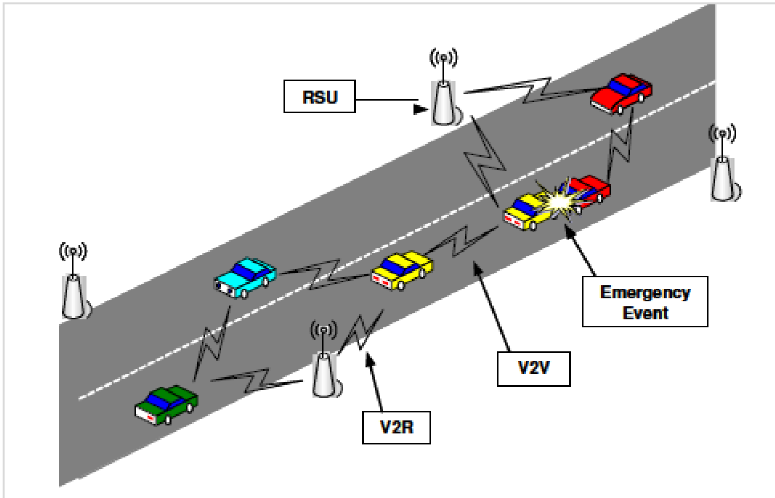

Figure 2.1 Vehicles to Roadside Communication Scenario [8]

The main functions of RSUs are to:

- Receive information from one OBU and forward it to another OBU, when extension of the communication range is required and the target nodes are out of sender range.
- Provide connectivity, to apply safety and warning applications.
- Operate as an information source, by sending information to vehicles using I2V communication (for example working zones and warning applications).

c) On-Board Unit (OBU):

OBUs are devices within vehicles that act as embedded systems for inter-vehicle communications [20]. They manage all the processes involved in collecting data from a vehicle's sensors, and collect data from other vehicles on the road. Information from external data sources is also collected via the RSU (for example TMC or the Internet using the IEEE 802.11p network device) [17, 20].

The OBU comprises the following components [19]:

- ***Resource Command Processor (RCP)***; this is responsible for executing commands and returning the appropriate responses to the resource manager (RM) in the RSU.
- ***Read/Write memory***; used to store and retrieve information.
- ***User Interface (UI)***; the OBU may provide visual displays, such as coloured light emitting diodes (LEDs), a buzzer to alert the user with an audible signal or/and text message display.

Several functions are provided by the OBU, including application, runtime environment, vehicle location and positioning, vehicle security and privacy, communication functions and interfacing with other nodes in range.

2.2.2 Communication scenarios in VANET

There are three main scenarios for VANET communication: in-vehicle, inter-vehicle and roadside vehicle communications, as shown in Figure 2.2.

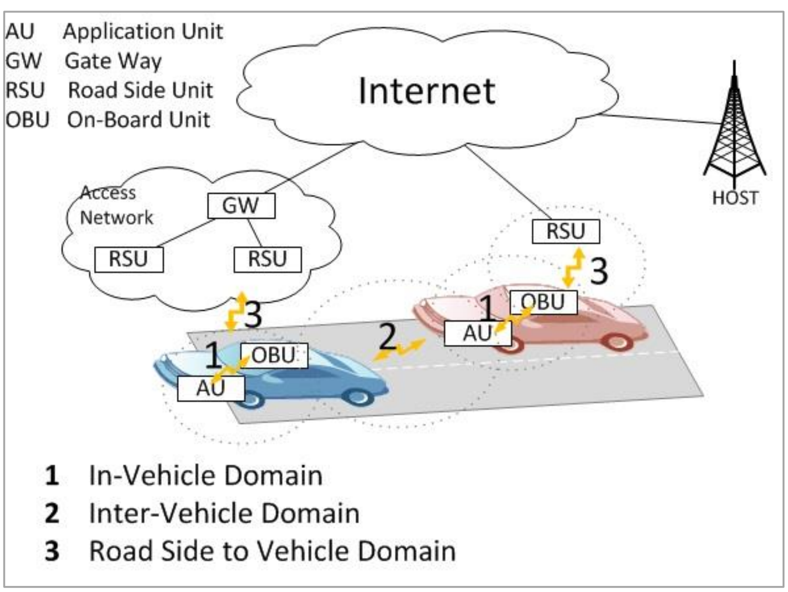

Figure 2.2 Types of communication in VANET

1) ***In-vehicle domain:*** This type of communication works inside the vehicle and is used to exchange information between OBUs and AUs. This connection can be wired or wireless (i.e. Bluetooth and Infrared). This type of communication system is essential for ensuring the integration of all the electronic devices inside the vehicle [21].
2) ***Inter-vehicle domain:*** In this type of communication, the vehicle can connect directly to all vehicles in range, without any need for an RSU. This form of communication is referred to as a vehicle-to-vehicle (V2V) communication scenario and is the most widely used type in intelligent transport systems (ITS). It is useful on roads, where RSUs are not available for reasons such as cost or complications affecting installation [21].
3) ***Roadside to vehicle domain:*** Roadside-to-vehicle domain, also termed Vehicle-to-Infrastructure Communication (V2I), is responsible for coordinating connections between nodes on the road and for ensuring that services are provided to all the nodes in the range [22]. The RSU acts as a coordinator controlling the negotiation processes and setting up the connection between infrastructures and mobile vehicles.

2.2.3 Wireless Access Technologies in VANET

VANET environments do not generally rely on a fixed infrastructure for the communication and dissemination of information; they operate in a highly dynamic environment. Therefore, there are three main categories of wireless technology that can achieve VANET communication: *cellular, WLAN/WiMax and hybrid wireless networks*, as shown in Figure 2.3. The Internet can be used to collect information from roads and traffic (for example routing, digital maps and advertising) [23, 24].

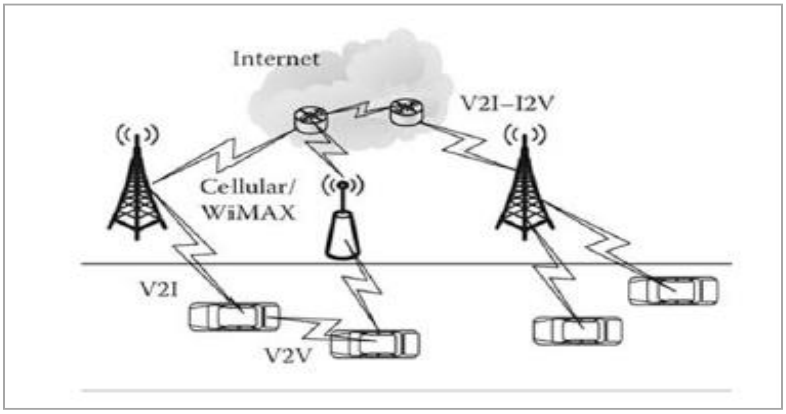

Figure 2.3 Wireless Access technologies in VANET

Wireless access technology functions as the main communication method between the nodes on the road. These technologies serve V2V, V2I and I2V connections and are important due to their reliability when delivering data to the destination; they communicate without delay considering important variables in driver and passenger safety applications [23]. VANET safety applications require fast and reliable communication between nodes. The most suitable and reliable wireless access technology, for improving VANET safety applications, is Dedicated Short-Range Communications/Wireless Access in Vehicular Environments Dedicated short-range communications (DSRC/WAVE).

Many researchers analysing DSRC communication, consider its reliability over DSRC-based Vehicle Safety Communication Applications (VSCA) in various environments (i.e. open field traffic environment, freeway traffic environment, etc.) The analysis shows that DSRC wireless communication provide an adequate degree of communication reliability in all traffic environments, and that packet drops do not occur even in a complex freeway traffic environment [22, 25, 26]. By incorporating appropriate estimation algorithms into the VSCA, the design of neighbour vehicle status information can improve the overall reliability of VSCA in order to provide satisfactory service to end users. Moreover an analytical model has been developed to relate DSRC communication reliability and VSCA reliability [27]. Another study, described in

[28], indicates that DSRC is better than 802.11a communication exhibits in terms of packet error rates; consequently it delivers higher channel capacity compared to 802.11a. The implementation of DSRC prototype systems and the field trials in [29] prove that real time constraints are met and that complex transactions can be completed during passage through the communication zone at regular driving speeds.

2.2.4 VANET Characteristics

VANET is considered to be a particular type of MANET, although characteristics differ between the two. Several factors are unique to VANET, including the following:

- Mobility and higher speeds (over the road speed limit) of nodes, resulting in fast changes to the network topology. Vehicles are restricted to moving using roads and must adhere to traffic rules, road conditions, traffic lights, speed limits and traffic conditions [30, 31].
- A number of mobility patterns have been observed and some statistical mobility models for VANET have also been designed [32].
- Drivers can respond according to the message contents received across the network; this will affect the behaviour of the driver and be reflected in the network topology [30, 31].
- All relevant information about vehicles, such as their current position, movement direction, current velocity, city map and planned movement trajectory of VANET nodes is increasingly becoming available, as more vehicles are now equipped with GPS devices and navigation systems [32].
- VANETs do not encounter the problem of energy limitation, due to their higher computational power and practically unlimited memory capacity, when compared to some other ad-hoc networks (particularly sensor networks) [33].

- VANETs may be either very large-scale networks or small networks, such as those on highways [34].
- A variety of services and safety applications are provided by VANET using V2V and V2I communication. VANETs provide multi-hop and single-hop communication between nodes using specific protocols [34].

2.2.6 VANET Applications and Services

VANET applications attempt to solve different problems and each application has specific communication requirements. Figure 2.4 shows how applications may be grouped according to their general aims; such as applications that provide information services or individual motion control.

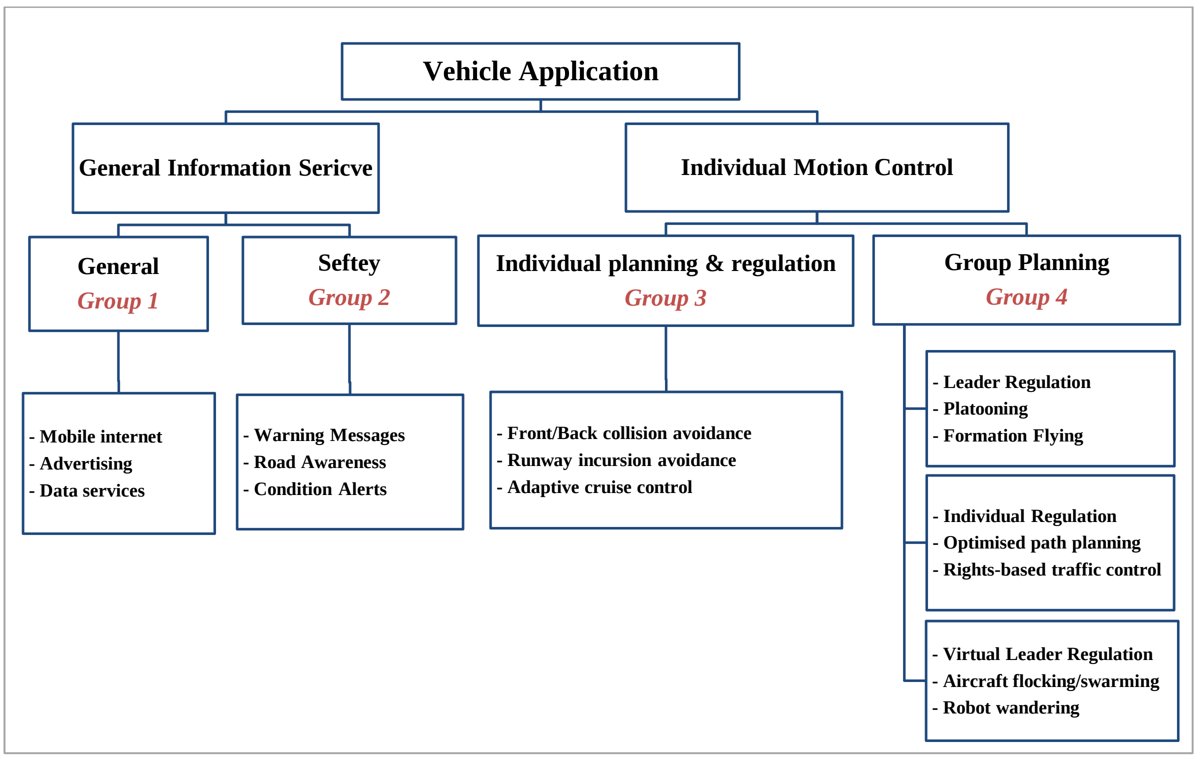

Figure 2.4 A taxonomy for VANET applications [10]

As shown in Figure 2.4, the vehicle application can be classified into two main classes; a General Information Service class and Individual Motion Control class.

The ***first class (General Information Service)*** represents all vehicle applications that are responsible for sending signals to a set of vehicles within a region, or throughout the network, on a public road; this relates to the provision of road services, advertising relating to cafés, shops and shopping centres on the road, warning messages about something happening or rules and conditions of the road. This covers information, the delay or loss of which does not compromise safety or render the application useless. The ***second class (Individual Motion Control)*** represents vehicle applications that have individual motion control and the capability to control the vehicle and change the decision of the driver, to improve the driver's safety on the road.

Below is a brief explanation about each group as shown in Figure 2.4:

Group 1: ***General Information Service***

The applications in this group are not involved in vehicle safety; therefore, they require a low communication overhead and a high information delivery ratio, owing to its main purpose being to multicast or broadcast a set of messages to a group of nodes within communication range. Vehicles equipped with sensors or a GPS can collect and analyse data about the environment and share this information with other vehicles (i.e. Mobile advertisement service [35], Roadside Service Finder Service [36], Internet Access Service [36, 37].

Group 2: ***Safety Information Service***

Safety applications are considered to be more sensitive to information latency than quantity of information; therefore, information latency will usually increase with source or receiver distance increasing, due to the physically interactive nature of these applications. This type of application is responsible for issuing safety alerts or emergency warnings related to prospective risks on the road, such as abnormal vehicle behaviour or poor road conditions. These warning messages need to identify the location and motion state, such as acceleration and velocity, of vehicles to send early warnings to nearby vehicles about an abnormal and suddenly arising situation. Some examples of this group of warning messages are: Traffic Signal Warning Application [38, 39], Stop Sign Warning Application [8, 40], Intersection Collision Warning Application, Pedestrian Crossing Intersection Application, Emergency Vehicle Warning Application, Traffic Signal Pre-emption Emergency vehicles, Curve Warning Application, Low Parking and Bridge Warning Application, Wrong Way Driver Warning Application, Work zone Warning and In-vehicle signage application [41].

Group 3: ***Individual planning and regulation of Motion Control***

Applications in this group use information such as current position, velocity and acceleration between vehicles to control a vehicle's position, velocity, acceleration, bearing and actuators to avoid collisions. All the information that is used to control the vehicle must be of low latency, extrapolations should be made to assess time-to-collision, conditionally issuing warning alerts and collision avoidance plans. This may include motion and actuator state broadcasts for vehicle collision avoidance and look-ahead data to improve the performance of adaptive cruise control [10]. Pre-crash models and applications are considered in this group because these types of applications are critical and need low latency and reliability for sending/receiving messages between vehicles in range to avoid collisions and to warn about, or process preventative action. Below are some examples of the applications in this group:

- ***Cooperative forward collision warning application***: This application enables vehicles travelling ahead of a vehicle that has crashed to avoid collision [10].

- ***Vehicle-based road conditions warning application***: This application notifies vehicles on a road about current conditions. This application collects sufficient information about the road's conditions through the use of sensors.
- ***Emergency electronic brake lights (EEBL)***: This safety application is considered to be the first application using only a V2V communication type. The application issues a warning if vehicles ahead have to brake suddenly.
- ***Lane change warning application***: This application warns vehicles attempting to change lane of the associated danger level.
- ***Blind spot warning application***: This application alerts a driver about whether there is a vehicle in his or her blind spot.
- ***Cooperative collision warning application***: This provides predictions regarding the probability of an accident occurring. Information can be collected from surrounding vehicles using the V2V communication system. This information includes direction, position, acceleration, yaw-rate and velocity, and is processed by the vehicle to obtain a prediction of accident occurrence.
- ***Cooperative adaptive cruise control***: This application works by adjusting vehicle speed relative to vehicles in front and behind using V2V communication.
- ***Road condition warning application***: This application is responsible for sending alerts on road conditions to prevent accidents.
- ***Pre-crash sensing application***: This application predicts the on-going situation and determines the type of crash and the danger level that may occur, by taking advantage of the information collected from sensors and data received from other vehicles, using V2V communication. This application improves the driver's safety.

<u>Group 4:</u> ***Group Planning Motion Control***

In this category, vehicles are grouped to facilitate complementary trajectory planning and may couple their motion with one another and vehicles that remain within the same group for a long period of time (i.e. minutes, hours). These organised vehicles may need to exchange control and status information to support the interaction between them. In addition to planning, numerous motion regulation architectures requiring low latency data exchange, namely individual, leader, and virtual leader regulations, are considered. In these scheme, vehicles work together to achieve real-time motion regulation, which may proceed on the indication of a preceding vehicle or be distributed entirely depending on the application's needs [10]. Some examples of the applications of this group are:

- ***Leader Regulation***: In this scheme, one vehicle broadcasts motion references and commands to the all vehicles in the group [42, 43].
- ***Individual Regulation***: VANET may be used to plan the global optimisation of vehicles that encounter one another, but which do not necessarily have the same destination or follow the same route [44].
- ***Virtual Leader Regulation***: This is a virtual or distributed leader model. This type of application can be used in many VANET communications to carry out complex group manoeuvres, such as flocking or swarming, while maintaining safe distances between vehicles [10].

The proposed pre-crash system in this thesis is considered as an individual planning and regulation group (Group 3) of the VANET Applications and Services, which have the ability to predict the crash likelihood and severity level.

2.3 Context Aware System (CAS) Overview

CAS were first presented in 1992 in the Active Badge Location System project [45]. CAS aim to increase the effectiveness and usability of contextual information as sensed from the surrounding environment.

Deys and Abowd [15] define CAS as: "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task." CAS has been used by different disciplines in the Information Communication Technology (ICT) field, such as web services, Distributed Artificial Intelligence, Artificial Intelligence Traffic System (AITS), Natural Language Processing, Knowledge Representation and Reasoning (KRR), Human Computer Interaction (HCI), information retrieval and computer security and privacy [46].

2.3.1 Context Definitions

The context refers to any information or fact sensed or gathered from the surrounding environment. It is important because system behaviours must adapt or change in accordance with available information, to maximise user effectiveness, such as comfort and safety, and to describe situations involving real-time activity. Many definitions of context exist; some researchers define it according to type (for example time, location, device or users) [47, 48]; others according to sources (e.g. memory, power supply and computing power) [11, 48] or current situation (at a particular location or time-period) [49, 50]. The most general and comprehensive definition of context is:

"Any information that can be used to characterise the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves." [51]

Context information is a simple classification of information based on entity type; the function of entities can be augmented. This classification helps system developers to understand and generate additional pieces of context from one context, increasing confidence in the characterisation and evaluation of the current situation.

The four main classes of context information are [5]:

a) ***Identity:*** each context should have a unique identifier, to distinguish one context from another, and to improve usability and context derivation.

b) ***Location:*** this indicates the position of an entity. This information can be expanded to include more context data related to the place, such as orientation, direction and elevation, and can be used to assume a relationship between all the entities in a proximal environment.

c) ***Status:*** this characteristic identifies the intrinsic and inherited characteristics of the sensed entity and some related attributes. For example, attributes related to a place entity include temperature, light and noise. In software and systems, this refers to attributes such as upload time for a CPU, the existence or state of files in a file system or the status of running queries.

d) ***Time:*** each entity has a historical information that helps increase context richness and helps predict future situations. It is also used in conjunction with other contexts or entities. Some systems use timestamps, time period, duration time and expiry time.

Zainol and Nakata [11] divided context identifiers into three main categories, as illustrated in Figure 2.5.

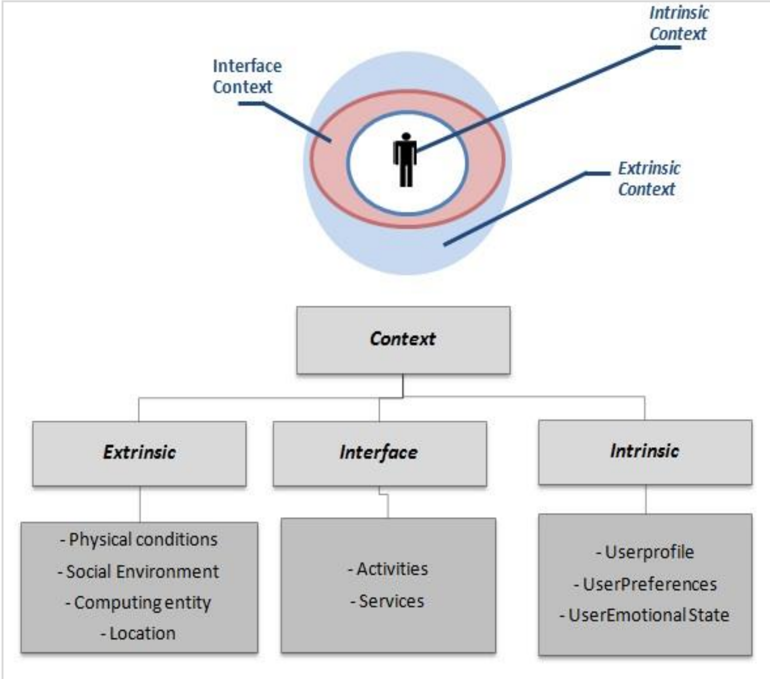

Figure 2.5 The layers of context categorisations [11]

- ***Extrinsic Context***, The outer layer of context categorisation, which represents the environment itself: physical conditions (e.g. light condition, weather condition, vehicle speed, day, road type, , road map and time), the social environment, computing entities (e.g. TMC, other vehicles position, and blind spot), and location (e.g. vehicle; direction, previous location, next location, current location and lane position) [11].
- ***Interface Context***, this interacts with the surrounding environment, representing user activities and services information and when tasks occur (for example driver activity, driver behaviour, service description like auto brake, airbag, hello message, and the service performance) [11].
- ***Intrinsic Context*** describes users' personal information and their emotional states and preferences, which they can provide directly (e.g. driver profile, vehicle profile and driver history).

The proposed system take in to account too many factors to reasoning and infer the results, these factors actual represent sensed context related to the vehicle itself, road, and environment factors. Each context has its identifier, location, status, and time.

2.3.2 Context Attributes

Each context atom can be represented and described as having a pair of attributes. The two most understandable are:

- ***Context type*** such as location, time or speed. This information is used in query arguments. The name of the context type should be unique, particularly in large software programmes. For example, the type position may belong to a vehicle or a pedestrian.
- ***Context value*** refers to the value of the raw data captured by a sensor. This value depends on the type of context and the sensor type, for example geographical coordinates, degrees Celsius or miles per hour.

2.3.3 Context Sensing

The word sensor refers to any source that can provide context information. Sensors can be [52]:

- ***Physical sensors*** that capture physical data; many are available. These hardware sensors are the most frequently used [53].
- ***Virtual sensors*** that capture context data; not from the physical environment or physical sensors, but from other software applications or services.
- ***Logical sensors*** that make use of the two sensor types described above. Physical and virtual sensors are used in conjunction with additional information from databases or other sources, to resolve higher level responsibilities.

CAS acquires context data in three different ways; these define the architecture of the system as follows [54, 55]:

- ***Direct sensor access***: In this type of architecture, the system has built-in sensors that sense and capture contexts directly. This type of system will acquire these contexts from the surrounding environment, and apply them to the application directly to obtain a reaction. This type of architecture has no layers and is therefore unsuitable for distributed systems, as it senses in a direct manner, leading to an inability to manage multiple sensor accesses simultaneously.
- ***Middleware infrastructure***: This is a layered architecture uses context encapsulation to conceal the physical raw data in low-level sensing hardware from other high level processing. This approach works as an interface to ease extensibility, and simplifies the reusability of contexts sensed from low-level layers.
- ***Context server***: This is a distributed system used to extend middleware architecture, to access context remotely and share it with multiple clients using a context server; thus, relieving clients of resource-demanding operations.

This may be useful if the device has some limitations in its processing power or storage space. The context server is design to manage these limitations, and to consider parameters such as appropriate protocols, network performance and quality of service.

2.3.4 Context Modelling and Reasoning

The modelling of context information is one of the first issues that need to be address after context information has been obtain. It is based on data structures and is responsible for presenting contextual information to the system in an understandable and meaningful form [56].

Context modelling is required to define and store collected data (context) in machine executable form, Some of these modelling methods are: Key-Value models, Mark-up scheme models, Graphical models, Object oriented models, Logic based models and Ontology-based models. These context models do not support reasoning about uncertain contextual information [57]; due to their reasoning limitations they are only able to define and store determinable (certain) contextual data, such as the temperature of the room, light, weight, etc. [54, 58]. They lack the ability to enact a reasoning mechanism [11]. The ability to act in some cases directly depends on certain information receiving low-level representation (for example; driver speed warning when the vehicle speed is above 120km/h).

The contextual information may be certain or uncertain; certain information means that the context-aware system (CAS) reacts or applies a service depending on one entity (context). This directly depends on an exact value without the need to check or analyse the context. This is considered low-level information. Uncertain information means that the CAS takes an action; applies a service or makes a decision depending on multiple entities (a combination of more than one context is used).

Context modelling must support reasoning about certain and uncertain information. However, in some cases the system cannot determine appropriate action directly; therefore, it is necessary to analyse and decide how to respond to uncertain information, and how best to capture more than one context and represent it at a high-level, using reasoning algorithms. For example when predicting whether a crash will happen or not, there are many variables that need to be known (i.e. if the vehicle speed high, if the weather is rainy, if the day is at a weekend, if it is a peak time of the day, what the traffic density is, what relative vehicle acceleration applies, distance to other objects, vehicle direction, road junction type, road speed limitations, road surface, etc.).

2.3.5 Reasoning about Uncertain Contextual Information

User situations may be considered as uncertain contextual information and cannot be directly acquired by sensors. This information needs to undergo some analysis if a high level of context information is to be obtained. Uncertain information cannot be captured directly by sensors, and the context may be incomplete, inaccurate or contain inexact information. For example, CAS reasoning cannot be judged by only accessing simple information about the vehicle, such as steering angle, speed or direction, and whether it is out of the driver’s control [57, 59]. To improve decision quality and heighten certainty, the system should sense the same context from more than one sensor; however, this is still not enough information to make correct decision; therefore, working under uncertainty by applying reasoning techniques to these contexts can insure the correct judgment is made effectively and accurately. This reasoning technique is used to obtain high-level contextual information from the information (low-level) sensed using reasoning techniques to drive and fuse information [12]. There are several reasoning techniques associated with uncertain context information, including:

- ***Fuzzy Logic:*** This method used to define semantic variables. For example, the crash type can be describe as crash vital, light, chain, rear, front, and side. Fuzzy logic is suitable for multi-sensor on, and can be used to describe context-subjective information and to resolve conflicts between contexts that are directly sensed. The concept informing this method is based on the creation of a new fuzzy set from two or more fuzzy sets. The problem with fuzzy logic is that it lacks the ability to perform reasoning about incomplete information [57, 58, 60] and its lack to support reasoning over time.

- ***Probabilistic Logic:*** This method of reasoning based on writing rules to follow to anticipate the probability of events. This method allows the architecture to predict and describe a high-level probabilistic context for any new situation using these rules [61]. However, it does not provide adequate expressive power to capture uncertainties and dependencies between variables, and to model the temporal aspects of the domain [57].
- ***Hidden Markov Models (HMM):*** A hidden Markov model is a statistical tool for representation probability distributed over a series of observations and events known as a Markov chain. HMMs have states and transitions, the states are hidden and only the outputs are visible [57, 58, 60]. The HMM represents the hidden state in terms of a single random variable.
- ***Dempster-Shafer Theory (DST):*** Also known as the theory of belief functions; this is a mathematical theory of evidence that is used to combine separate contexts to determine the probability of specific events. This theory is an undirected graph, therefore it is more complex to handle reasoning under uncertainty [57, 58].
- ***Neural Networks:*** neural networks are comprised of interconnected constituents known as neurons; they were design to mimic an extremely simplified model of the human brain when acting and performing tasks. They are appropriate for resolving problems that require a huge number of inputs mapping into small outputs. They use the neurons' capabilities to perform parallel and a non-linear computing [62-64]. The prediction capability of such a network is less accurate than other types of reasoning techniques, such as Bayesian networks [65], and the training of the network is usually slow [66].

- ***Bayesian Belief Networks (BBN):*** This method represents the distribution of joint probability and has two components: the directed acyclic graph and the variable conditional distribution (relationship between graph nodes). It may be used when reasoning about an uncertain situation [57, 61]. Static Bayesian Belief Networks (BBN) are used to represent systems with static property, working with a single time slice, which is unsuitable for systems with dynamic property, that require more than one time slice to represent it. The most important properties of this method are the ability to infer low-level context to high-level context and to learn (update) the values of the BBN.

  BBNs have become an effective tool to address uncertainty in artificially intelligent and knowledge based systems; they are attractive by their ability to encode joint probability distributions (JPD) efficiently, and to define a conditional distributions table (CDT) accurately. Moreover, BBNs had significantly developed to understand any complex model and represent it; they also have the ability to calculate missing values.

- ***Dynamic Bayesian Network (DBN):*** Dynamic Bayesian Networks are a set of Static Bayesian Networks interconnected by sequential time slices. DBN is use to solve problems, which have a dynamic property and change over time. A DBN is suitable for prediction, which has the capability to work under the conditions of changing events over time.

  This method is deployed in our system as a reasoning algorithm, in the context-aware system reasoning phase [57, 58, 60]. The full details of this will shown in chapter three of this thesis.

2.3.6 Advantages of Dynamic Bayesian Networks over other representations

Recently, DBNs have become an effective tool used to deal with uncertainty in Knowledge Based and Artificial Intelligent systems. The DBN graphical structure used to develop and understand any complex model and represent it in easy way, it can simply represent relationships between domains and attributes [57, 58, 67-70].

In addition, DBNs have the advantage of the ability to:

- Integrate and incorporate prior knowledge.
- Fill out the missing values and incomplete data sets.
- Infer low-level contextual information to a high-level context.
- Operate efficiently as a decision making model (such as choose the best next action, movement or make decision, etc.).
- Learn about the relationships between network parameters and network structure.
- Combine expert knowledge and datasets.
- Avoid the over fitting of data during learning.
- Model the causal relationships between network parameters.
- Support systems that have a dynamic nature (change over time).

DBNs might use to demonstrate and reason about uncertain situations and provide various reasoning concurrently, as presented to solve problems which have a dynamic property, and which are suitable for crash prediction. This is the best method for supporting work in conditions where events are changing over time.

In addition to the DBN advantages, DBN has some disadvantages also; the first problem is the inference, which has to be unrolled to represent the dependencies between DBN nodes from the different time slices and the conditional relation between nodes as well as influence that evidence nodes have on hidden nodes. The unrolling complexity is increase by increasing number of nodes. Moreover, Dealing with parameter learning in DBN can be difficult to enhance the learning of any DBN.

2.3.7 Context Aware Systems (CAS)

The CAS is responsible for adapting operations to the current context, ensuring it is more effective and usable relative to the environmental context [12, 71]. CAS has the ability to sense, detect, interpret and respond to the surrounding context and use this information to serve and provide for the user's needs [49, 72, 73]. The design and application of CAS depends on system requirements and the services required by the designer [11].

CAS provide information and services to users in the following ways [74, 75]:

a) Proximate Selection CAS, this CAS can provide sensed information or support any changes in the information distributed to users, or suggest a set of services to help users to select a suitable action based on the current situation [76], for example; Tour guide [46] and Network maintenance [77, 78].

b) Automatic Contextual Reconfiguration, this CAS is responsible for the reconfiguration of any changing of the system (adding or removing components), or altering connection and communication channels between components. Any changes to this class require the adding, loading, and remodelling of hardware components [76]. Examples include teleporting [79], intelligent control of audio visuals [80], and Office environment control [81], Call forwarding [45].

c) Contextual Information and Commands, This CAS generates and produces different results and commands (actions) according to the contextual information [79], for example automatic integration of user services [62, 82, 83] and in/outboard [63].

d) Context Triggered Actions, This CAS is responsible for tagging the collected context information and then relating it to relevant context information. This works with If-Then rules, which used to specify how CAS should adjusted. These actions are automatically applied according to previously specified rules [46], for example capture of a classroom lecture [46], tour guide [84], paging and reminders [73, 84, 85], conference capture and tour guide [86], fieldwork data collection [49, 72, 87], in/out board capture of serendipitous meetings [63] and virtual post-it notes [88].

The information were collected then modelled using a context modelling method and a specific reasoning technique is use to obtain high-level information as required. CAS may provide several applications and services, depending on the information processed. CAS accommodates and autonomously responds to any changes in environmental contextual information [11].

In this thesis, the Context Triggered Actions CAS method is uses to represent our system, which works to tag the context information, collected and then relate it to relevant context information. Moreover, it works with cause rules, which used to specify how CAS should adjusted and determined.

2.3.8 Context Aware System Architecture

Several architectures have proposed for context-aware systems. The architecture may depend on the location of sensors (local or remote), the number of possible users and the availability of resources. Context acquisition predefines the style of the system, and is therefore, considered a very important issue that must take into account before the design of any CAS.

In the next section an explanation to, the *Layered Conceptual Framework*, which considered as the main, most general and famous context aware system architecture. An in depth description will be provided to explain the main concepts behind CASs and to explore how context data is collected and analysed to provide services to the user. Our pre-crash system architecture built upon this framework, as will show in chapter 3.

The ***Layered Conceptual Framework:*** A general layered conceptual framework for Context Aware Systems containing five main layers, exists as shown in Figure 2.6 [55]. The layered conceptual architecture augments layers for the detection and use of contexts by adding interpretation and reasoning functionality [5, 61].

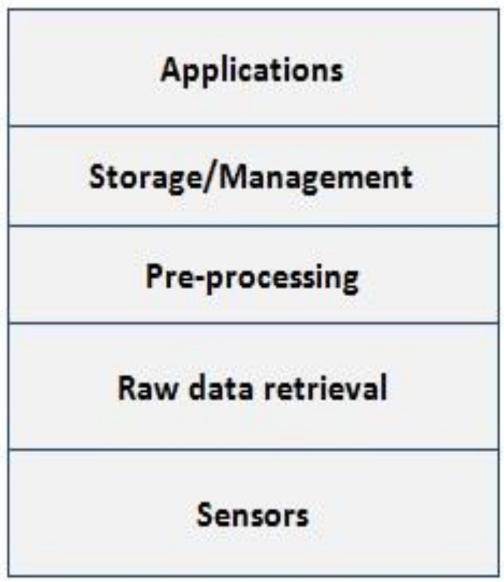

Figure 2.6 Layered conceptual framework for CAS [12]

This framework contains the following layers:

1. ***Sensor layer****:* This layer captures and collects raw data from the surrounding environment using different type of sensors.
2. ***Raw Data Retrieval Layer****:* This layer retrieves raw context from the sensing layer (physical sensors) after sensing, capturing and collecting context data from virtual and logical sensors. This layer works in a low-level manner and often assists the usability of software components. It is considered to be an interface between low-level and high-level layers in CAS; it is responsible for hiding the details of the physical layer, and helps make

the CAS components and devices (such as sensors) exchangeable (e.g. exchanging system components without any major configuration or modification in this layer or the higher layers).

3. ***Pre-processing layer**:* This layer is very important in CASs. Its main purpose is to obtain high-level contextual information by performing reasoning and interpreting low-level contextual information. Depending on the system design, this layer is not available in some CASs, which is required to work with certain information only, without the need for any reasoning or analysis for context information to take action based on the current situation. (For example, a speed alert in the vehicle does not need any analysis or reasoning to warn the driver about being over the limit speed, in this situation only a speed sensor needed, and alerts would sent when the speed value exceeds the road limit). This layer needed when the process or the decision depends on more than one context from different sources and requires analysis, to make the right decision. This layer deals with high-level context information; any context atom should combined to a high-level valued and accurate at the same time (for example when it is necessary to describe a road situation depending on multi-contexts such as weather status, road conditions, road density, and vehicle velocity). All these contexts used to describe the situation in an exact moment according to real-time requirements. This layer also solves the problem of conflicting contexts, due to the use of more than one sensing source, using additional data to support the correct decision, such as time stamps (more details about reasoning method and their techniques will discussed in section 2.4.5).
4. ***Storage and Management**:* This layer is responsible for organising and gathering, the captured and sensed contexts so it can used by the client or the server via a public interface.

This layer is useful for supporting the pre-processing layer, to predict future behaviour or situations using the context history. The client might work in two ways: a synchronous manner in which a client asks the server (by a request and response communication method) about any changes that occur remotely, or asynchronous, in which each client works according to a subscription if interested. An asynchronous way is more suitable for rapidly changing environments and has a high mobility.

5. ***Application layer:*** this layer is responsible to implemented and applied the actions and reactions of the CAS. It sometimes acts as an input sub layer, in which information are requested remotely from the server in a synchronised way, as mentioned in the storage and management layer. The actual reaction to different events and context-instances implemented here.

2.4 Crash Prediction Models Related Works

Several researchers have examined a wide range of methods for predicting crashes, reflecting the importance of crash predication systems within the ITS. Some have attempted to focus solely on the driver and others on the vehicle; none has focused on the combination of the contributory factors on the vehicle, driver and environment. As shown in Figure 2.7.

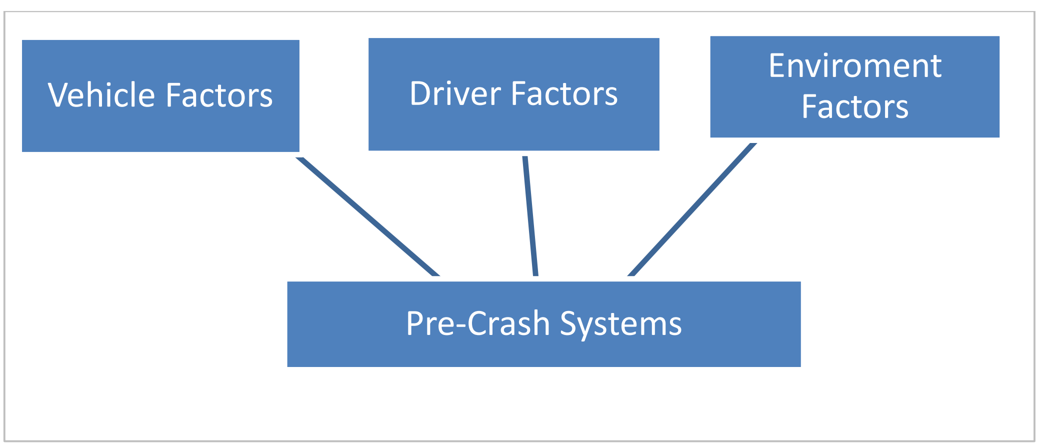

Figure 2.7 Pre-crash Systems classified according contributory factors that affect it.

A) Pre-crash Systems that are affected by Vehicle Factors Only:

This class of systems detects the probability of an accident based solely on the vehicle, which will inevitably lead to an inaccurate prediction.

Tokoro et al. [89] proposed a pre-crash safety system utilising adaptive cruise control (ACC). This system aims to enhance vehicle safety and reduce the injuries caused by collisions, by activating a pre-crash seat belt and pre-crash brake to assist in the event of an unavoidable accident. The author has developed millimetre wave radar as well as a new signal-processing algorithm in order to increase the accuracy of the ACC system. When the electronic control unit (ECU) detects an incoming obstacle based on the time to crash (TTC) and detects that a collision is unavoidable, the PCS will activate a seat belt and brake assessment. However, this system acquires information about the vehicle only, leading to inaccurate crash predictions. Moreover, the system works only in the case of unavoidable crashes, which is insufficient, as the aim of this work is to predict a crash early enough to avoid injuries.

Skutek et al. [90] developed a pre-crash system based on short-range radar. The aim was to design a pre-crash application to work in all weather conditions. The Kalman-Filter approach was use to fuse the information collected by sensors to detect the position of objects using distance and velocity information.

Skutek et al. [91] extended their work, as proposed in [90], in to add a laser scanner with short range radars to acquire additional necessary information. The information acquired included vehicle location, moving directory and velocity. The information was fuse using a grid fusing approach in order to guarantee high quality and accurate information about the ahead objects. Then the authors added new sensors to their system and applied an information fusing technique.

Cho, et al. [92] deployed a collision mitigation system (CMS) using a pre-crash algorithm and a variety of high technology sensors to calculate the possibility of a collision, the system takes into account the wheel speed, steering angle, yaw-rate, lateral acceleration, side-slip angle, longitudinal and lateral velocity and acceleration, as an input to the system. The pre-crash algorithms used to enhance the airbag deployment performance by calculating the crash possibility, time to crash (TTC) and crash type via estimated information. An original method for crash type decision making were proposed to enhance airbag deployment, which involved defining a possible crash zone and lowering the crash algorithm threshold using a bicycle model that had been modified.

Radzi et al. [93] proposed a collision detection algorithm for Unmanned Aerial Vehicle (UAV) systems to determine whether a collision between two moving UAVs was about to happen. Using the parametric form, an interception point was proposed and the difference in time taken for UAVs to reach the interception point was investigated. A collision detection algorithm was developed to predict the likelihood of collision between two moving UAVs. Parametric theorem methods (PTM) were used in collision detection algorithms, and consisted of: equation of a line to detect the collision in the future, interception point of the UAV, and collision detection algorithm to a predict crashes at an earlier time. The algorithm in the system was based on the vehicle's initial position point, the estimated final point and the velocity of all the UAVs.

Lindner and Wanielik [94] proposed new sensors, such as three-dimensional measuring multilayer laser scanner systems, that could deliver the necessary environmental information for these systems. Fusing of multiple sensors, such as laser and radar systems, was employed to significantly improve the reliability level of the automotive safety applications. An approach for the fusion of multilayer laser scanner (Lidar) data with radar measurements for the detection of vehicles in on road environments was used. The Lidar data was processed using a new three-dimensional occupancy grid. An example of the fusion of Lidar and radar information for a pre-crash application was presented.

Son et al. [95] developed a new measure for crash risk, called "unsafe following condition (UFC)", to estimate the possibility of a traffic crash using individual vehicular information. The UFC measure was proposed as a means of predicting the crash potential between two consecutive vehicles. This information was applied to basic sections along the interstate highways in Virginia, USA. Individual vehicular data and crash data was then used to develop crash prediction models. Six count data models, including zero-hurdle models, were compared to determine the best model type.

Eigner and Lutz [96] examined context collision avoidance applications in VANET, which were modelled using ontology web language OWL to express the context model and facilitate automated reasoning. This formalised the situation as a graph of a non-linear system to calculate the possibility of collision, by checking whether there was any overlap between the rectangles representing the vehicles. This model depends on data relating to the vehicle, such as: length, width, initial speed, acceleration and deceleration, All this data was collected from the OWL. The Fourier Motzkin Elimination was used in the model to calculate the time to impact and the overlapping between rectangles.

Karlsson et al. [97] developed a collision avoidance system using late braking to avoid or mitigate any collision. A Bayesian approach with implementation of an extended Kalman filter (EKF) method and a particle filter approach were used to solve the tracking problem and to make decisions. Comparisons were made between the two filters for different sensor noise distributions, using the Monte Carlo simulation method. The braking decision was based on a statistical hypothesis test, in which the collision risk was measured in terms of required acceleration to avoid collision. This model considered the state vector relative to position, velocity, direction and distance, to calculate the hypothesis node probability.

Ulrich Lages [98] designed a collision avoidance system to avoid fixed obstacles, using a fuzzy logic controller based on 72 avoidance manoeuvres. The system was designed to detect fixed obstacles solely by taking into account information about the vehicle.

Moritz [99] designed a present threshold algorithm to sense the crash before it happens using a radar sensor to detect the virtual barrier and detect the maximum and minimum detecting range required for data transmission, and actuator reaction times. The parameters used in this algorithm were closing velocity, barrier stiffness, mass, stiffness of own vehicle, and mass of own vehicle.

B) Pre-Crash Systems that are affected by Vehicle and Driver Factors:

This class of systems makes a decision based on the likelihood a vehicle will crash, according to driver factors, ignoring the environmental effects on crash likelihood.

Gnanamurth et al. [100] designed a middleware architectural model to provide safety in “high-speed” vehicles via a VANET network. This, InVANET system, was analysed for different situations, such as (a) high road traffic intensity; (b) slow-moving vehicles; (c) abnormal vehicle failure; and (d) drunk-driving. Different sensors were used to form a VANET network in addition to a vehicle communication structure. An experimental test-drive was carried out on national roads on which nine cars were used for 25 driving hours. It was concluded that the InVANET network experienced minimal delays when communicating with multiple vehicular nodes, as compared to other networks.

C) Pre-Crash Systems affected by Vehicle and Environmental Factors:

This class of systems depended on the crash prediction element of vehicle and environment factors and ignored the effect of driver factors.

Abdel-Aty and Haleem [101] developed multivariate adaptive regression splines (MARS) to predict a vehicle's crash angles. The MARS model is promising in terms of prediction and does not suffer from complexity of interpretation. The negative binomial (NB) model was compared with the MARS model, using extensive data collected at unsignalled intersections in Florida. Two models for angle crash frequency were estimated, one at three-legged and the other at four-legged unsignalled intersections. Crash frequency was examined as a continuous response variable for fitting a MARS model, by considering the natural logarithm of crash frequency. Finally, the MARS model was combined and examined with other machine learning techniques, such as random forest. It was concluded that the MARS model is an efficient technique for predicting crashes, in particular, angle crashes at unsignalled intersections.

Salim et al. [102] developed a context-aware system for collision warning and avoidance systems by proposing the use of a ubiquitous data-mining-based layered agent (UDMLA) model, engineered to support vehicle safety. This system may be applied at any type of intersection and integrates multi-agent systems with ubiquitous data mining designs. Sensor technologies were used for environmental perception (infrared sensing, video and camera image perception, LIDAR/RADAR sensors, gyro sensors sensing vehicle motion and acceleration, and inertial sensors such as tachometers and speedometers). An XML format was used for context modelling, and processing-sensory data obtained through mathematical algorithms, that resulted in a virtual understanding of the vehicle environment, such as the path and position of vulnerable road users relative to other vehicles and road infrastructures. Both [43, 103] ignored environmental factors, reducing the accuracy of the models.

Hsu et al. [104] developed a system for remote warning for vehicle collision avoidance using DSRC and GPS to transmit information and the position of the vehicles on the road. The proposed system used a mathematical model of conflict detection to calculate the collision time supported by geographic ellipsoidal coordinates and north-east-down navigation analysis in vehicle utilisation.

Caliendo et al. [105] developed a crash prediction model for a four-lane median-divided Italian motorway on the basis of an accident dataset extracted from Motorway Management Agency reports. The model depends on traffic flow, infrastructure geometry, pavement surface and rainfall variables and uses a statistical method to predict crashes.

2.5 Summary of the related work:

None of researchers designing crash prediction systems were affected by the three main contributory factors on the same system; vehicle, driver, and environment. This may lead to a limited condition, lack in accuracy, and confused understanding of the situation. The next paragraph clarifies the lacks and limitations in the systems reviewed in the literature review.

Limitation in conditions: all the above systems predict crashes without taking into account all the contributory factors in combination to describe the real situation; consequently crash predictions are inaccurate.

Estimated information: in [86, 87, 89, 95] models are used to generate pre-crash information using estimating information; this is very dangerous in such safety application, where accuracy and reliability are crucial factors.

The Pre-crash system will attempt to work under all weather conditions and take into account driver, vehicle and environmental factors. This can be achieved using Context-aware System (CAS), to collect contextual information about contributory factors. The adaptive Hello message is used to collect contextual information from the RSUs and TMC. Our system is also intended to have the capacity to make decisions utilising the DBN and dynamic context knowledge based foundation; which will allow for the analysis of collision patterns from traffic data that have been used to collect historical information about collisions on the roads, the driver and environment. Dependence on one sensor alone might lead to inaccurate values and subsequently the wrong decision being taken, therefore, adding more nodes and different sensors will be used to improve prediction reliability and accuracy rates.

CAS can help to achieve the goal of collecting and analysing the available information; it should be reliable, adaptive and have the ability to perform both modelling and reasoning about uncertainty to improve driver safety.

General Middleware context architecture was designed and the DBN method was utilised for reasoning, to improve the capacity for assessing new high-level contextual information from a low-level context. This method is commonly used in this field of reasoning and is a very powerful method for representing and reasoning in uncertainty [57, 106], as shown in section 2.4.6. DBN is responsible for reasoning and prediction, which is especially difficult to model using other reasoning methods. Machine learning was used with a foundation of dynamic contextual knowledge, to add the facility of learning to the proposed pre-crash system.

2.6 Summary

In this chapter, a comprehensive overview was presented of VANET and its architecture, characteristics, wireless communication, challenges and applications. A complete presentation illustrated CAS and its definitions, attributes, sensing, modelling, reasoning with uncertainty and the algorithms utilised as reasoning methods for high-level contextual information, characteristics and architecture. An investigation into reasoning techniques was presented, as was a brief explanation of why the proposed system utilised the DBN. Moreover, some related works were presented here and grouped according to those factors that affect their decisions regarding crash prediction. In addition, a critical investigation of these frameworks and models was presented with attendant weaknesses, and lacking decision making in comparison to the proposed system.

Chapter 3 - Preliminaries

Chapter Objectives:

1. Define the aim of the pre-crash system and specify the objectives that need to achieve.
2. Present and describe the main aspects of the proposed pre-crash system.
3. Overview of Bayesian Belief Networks and its functionalities.
4. Present Bayesian networks and their value.
5. Explain the existing DBN software packages and software justification.

3.1 Introduction

Pre-crash systems are a key technology that will have far-reaching implications for future mass transportation. Therefore, the topic of this thesis is very important because such systems are directly link to the lives of both the drivers and passengers. At the present time, safety applications are integrated into vehicles, although this evidently requires more improvement in order to prevent road crashes, to reduce the number of fatalities and to save people's lives.

Accordingly, governments, vehicle manufacturers and researchers are seeking to find and develop new technologies and systems. All these systems depend on the accuracy of the data captured from the surrounding environment and from other vehicles on the road. Nevertheless, to date no system has emerged that can reliably control all the potentially dangerous situations that might arise on the road (signed traffic, unsigned traffic, highways, curve roads, road intersections, etc.). Thus, most researchers choose to focus on developing systems to address certain restrictions only; meaning existing systems are unsuited to widespread application, and cannot cover all kind of roads and situations.

The DBN reasoning technique is used in our system to represent the system’s network graph; nodes (system’s variables), relations (directed arcs), and infer a network to find the hypothesis nodes’ values when some evidence is given.

An investigation of BBN software packages was undertake for the purpose of comparison; GeNIe 2.0 was chosen to deploy our pre-crash DBN.

3.2 Overview of Pre-Crash Systems

In VANET there are generally two types of safety applications, passive and active [92, 103].

- ***Passive safety application,*** this type of application deploys its action dependent on one condition or an indication of whether it is true, otherwise, the application does not introduce any action and remains OFF, such as; seat belt, over speed, etc.
- ***Active safety application,*** this type of application requires more than one indication to deploy its action and is active all the time either to make the driver aware of a hazardous situation or to aid the driver in avoiding a hazardous situation; such as, auto braking application, steering while changing application, etc.

Active safety applications are very complex and differ greatly from *passive safety applications* when developing pre-crash systems. Although early crash prediction applications used various technologies to detect objects, most pre-crash applications work with the assist of radar, GPS, cameras, etc. Pre-crash applications depend on a vehicle’s position, distance, speed, velocity, acceleration, and other related information about the surrounding environment, the driver and nearby vehicles. The application will not initiate action in normal situations, but it is immediately triggered to warn or avoid a crash if any sudden changes occur in those factors that could cause a collision, and crash likelihood is high enough to be sure about it [92, 103].

3.3 Overview of Pre-crash System Elements

At present, pre-crash systems are part of the standard safety equipment for modern vehicles [17], especially those used to enhance the safety of drivers and passengers in the event of a crash. Pre-crash systems must evaluated accurately at a predefined time before the driver encounters a hazardous situation. It is difficult to predict crashes from a single perspective; i.e. focusing solely on the driver or the vehicle, as this may not describe the complete situation on the road. Since 2005, all the police forces in Great Britain have been reporting contributory factors as an integral part of the STATS19[1] collecting system. The contributory factors system has developed to provide insight into why and how road accidents occur. Contributory factors are those that identify the key actions and failures that can lead directly to actual impacts. Knowing of them may aid investigation into accident avoidance. Therefore, a proper decision were needed; this can be achieved by collecting more contextual data using different kinds of sensors, taking into consideration all the factors affecting everyday driving situations. The system proposed in this thesis uses real-time traffic and contextual information about contributory factors to predict crash likelihood and severity levels.

[1]STATS19 is a database in which police throughout Great Britain record details of road accidents that involve personal injury. The basic details of the people, vehicles and roads involved in these accidents are recorded. The modern STATS19 was established in 1949, and the current collection system became established in 1979 following a wide-ranging review. The current STATS19 form consists of an accident record, a vehicle record (for each vehicle) and a casualty record (for each casualty) [107] (2013, 27 July). *stats19 Help File*. Available: http://www.stats19.org.uk/.

These factors are categorised as having three main elements, with examples in each category [108-111]:

- **Vehicle:** speed, distance, lane position and vehicle defects.
- **Driver:** driver's age, driver's behaviour and driver's error or reaction.
- **Environment:** contains two groups:
  - Road types: dual carriageway, one way street, roundabout, etc.
  - Weather conditions: road surface conditions, traffic density and time.

Based on a UK DfT report [1], four levels of crash severity were considered: fatal, serious, slight and no crash, as shown in Table 3.1. Crash severity will directly related, but not limited, to some of the above factors. These factors have different stages of prevention at each crash level. This will be fully illustrated in Table 5.1 in Chapter 5 – page 88, the column of the main contributory factors in the below table shows each factor and how it can affect the crash.

Class No.	*Crash level*	*Definition*	*Main contributory factors*		
			*Vehicle context**	*Environmental context**	*Driver context**
1	**Fatal**	An accident in which at least one person is killed.	- Over speed limit - Sudden acceleration - Unsafe distance - Bad lane position	- Bad weather - Bad road surface - Bad time of the day	- Bad driving behaviour - Age - Gender
2	**Serious**	An accident in which at least one person is seriously injured but no one is killed.	- At speed limit - Sudden acceleration - Unsafe distance - Bad lane position	- Bad weather - Bad road surface - Bad time of the day	- Bad driving behaviour - Age - Gender
3	**Slight**	An accident in which at least one person is slightly injured but no one is killed or seriously injured.	- Below speed limit - Unsafe distance - Bad lane position	- Bad weather - Bad road surface - Bad time of the day	- Bad driving behaviour - Age - Gender
4	**No crash**	When no accident occurs.	- Stop - Safe distance - Good lane position	- Good weather. - Good road surface - Good time of the day	- Good driving behaviour - Age - Gender

Table 3.1 Crash levels and the contexts that may be responsible for a crash.

The (*) in Table 3.1 refer to samples of the values that might affect the decision to be in one of the mentioned crash severity levels. For example, if the diver drive his/her vehicle over the road speed limit, unsafe distance and in the same lane of the ahead one, that will lead to fatal crash for sure if the driver do not take any action, the environment factor might increase the crash possibility, i.e. bad weather (e.g. raining) will increase the crash likelihood, and if the driver has bad behaviour that also might led to increase the crash likelihood.

There are unlimited crash scenarios in the real world, and no comprehensive system can predict them all. According to the DfT [1], crashes can be classified into four main types:

- Back: the back side of the vehicle .
- Front: the front side of the vehicle.
- Nearside: the passenger side.
- Offside: The driver side.

The proposed system is capable of predicting back and front crash types only and their level of severity. In other words, the system is able to detect a crash in cases where the host vehicle and the vehicle ahead are in the same lane, in any road surface conditions, the nearside and offside crashes are not included in this thesis and the prediction of these type of crashes will done in the future.

3.4 Pre-Crash System Assumptions

It was assume in the Pre-Crash system developed for this thesis that:

- Each vehicle is equipped with a number of sensing technologies (the source of contextual data).
- Each vehicle on the road has wireless communication with other vehicles, to manage V2V and V2I using a DSRC/WAVE device.
- Each vehicle is provide with a built-in profile, showing the vehicle manufacturing details (for example vehicle size, weight, etc.).

- Each vehicle is provide with a built-in driver profile showing full details about the driver (for example age, gender, driving style, and driving history).
- Road information is manage and update by the Traffic Management Centre (TMC).

3.5 Pre-crash System technique

The Dynamic Bayesian network method was used to demonstrate and reason about uncertain situation and provide various reasoning types simultaneously. It is suitable for crash prediction and it is the best method with the capability to work under conditions where the system is changing over time. In this section, a depth representation of the DBN will introduced. DBN is commonly use in this field of reasoning and is a very powerful method for representation and reasoning under uncertainty; as shown in section 2.4.6.

3.5.1 Background of Bayesian Belief Networks (BBN)

The Bayesian Belief Network is a directed acyclic graph (DAG) comprising two types of components; a set of nodes which essentially represent the variables in BBN and a set of directed edges (arcs), which represent the relations between these variables. The purpose of BBN is to infer unobservable events from observed events [68]. Each variable is link to local probability distributions; the BBN is represent with a set of conditional independence assertions. BBN can be specified with Bayesian probability equations obtainable from the product of all conditional probability tables within the BBN [112, 113].

BBN are supplementary and helpful in cases where the values of some variables have given, (evidence), and where other values are not (hypotheses). In the real world most of the facts (variables) have continuous values, such as; Speed, velocity, temperature, and age, while the BBN works with distributed values and each variable must has its own CPT), therefore the BBN needs to be distributed; these continuous values make these values accessible to create a variable's CPT; as shown in Table 3.2. Also, the GeNIe software cannot works with continuous variables as will be shown in section 3.6.

	Node name (variable)	**Rules** [112]	**Values**
Continuous values	Age	$f(X_1, \dots, X_n) = f(\mathrm{X}_1) \prod_{i=2}^{n} f(\mathrm{X_i} \mid \mathrm{X}_1, \dots, \mathrm{X}_{\mathrm{i}-1}) \dots (3.1)$	Real numbers: i.e. 1.00,2.00,..,70.00, etc.
Distributed values	Age	$p(X_1, \dots, X_n) = p(\mathrm{X}_1) \prod_{i=2}^{n} p(\mathrm{X_i} \mid \mathrm{X}_1, \dots, \mathrm{X}_{\mathrm{i}-1}) \dots (3.2)$	States (sets): - Under 17 years - 18-30 years - 31-50 years - 51-70 years - Over 70 years

Table 3.2 Example of Age variable represents its values in continuous and distributed way

In the proposed system, all the variables have distributed values; therefore, there is no need to explain the distribution methods that fall beyond the thesis' scope.

3.5.1.1 BBN Probability Theory

The core framework of BBN theory is *Bayes' Theorem*; this is a theorem of probability theory, originally stated by the Thomas Bayes (1701–1761). It can be seen as a method for understanding how the probability that a theory is true can be affected by new observational evidence. This theorem was used to update beliefs and to illuminate the relationship between theory and evidence. This theorem is declared as follows:

$$p(X|e) = \frac{p(e|X)p(X)}{p(e)} \dots \dots (3.3)$$

Where:

- $p(X|e)$ is the posterior probability of hypothesis X, given evidence e.
- $p(e|X)$ is the probability of evidence e, given hypothesis X.
- $p(X)$ is the prior probability of hypothesis X.
- $p(e)$ is the prior probability of evidence e.

Equation 3.3 states that by observing evidence, the prior probability of hypothesis (X) can altered to a posterior probability.

The BBN theorem, is in fact obtained from the ***product rule*** (also known as the chain rule), which is obtained from a definition of conditional probability. This product rule is derive by writing Bayes' Theorem in the following form:

$$p(X,e) = p(X|e).p(e)$$

$$= p(e|X).p(X) \text{ ... ... } (3.4)$$

Successive applications of the product rule yield the chain rule:

$$p(\mathrm{X_1},\ldots,\mathrm{X_n}) = p(\mathrm{X_1},\ldots,\mathrm{X_{n-1}}).p(\mathrm{X_n}|\mathrm{X_1},\ldots,\mathrm{X_{n-1}})$$

$$= p(\mathrm{X_1},\ldots,\mathrm{X_{n-2}}) \cdot p(\mathrm{X_{n-1}}|\mathrm{X_1},\ldots,\mathrm{X_{n-2}}) \cdot p(\mathrm{X_n}|\mathrm{X_1},\ldots,\mathrm{X_{n-1}})$$

$$= p(\mathrm{X_1}) \cdot p(\mathrm{X_2}|\mathrm{X_1}) \cdot \cdot \cdot p(\mathrm{X_n}|\mathrm{X_1},\ldots,\mathrm{X_{n-1}})$$

$$= p(\mathrm{X_1}) \prod_{i=2}^{n} p(\mathrm{X_i}|\mathrm{X_1},\ldots,\mathrm{X_{i-1}}) \text{ ... ... } (3.5)$$

For every $\mathrm{X_i}$, there may be a subset $(\mathrm{X_i}|\mathrm{X_1},\ldots,\mathrm{X_{i-1}})$, such that $\mathrm{X_i}$and the subset are conditionally independent. This means that this subset can left out of the original set$(\mathrm{X_1},\ldots,\mathrm{X_{i-1}})$. When the subset is empty, X_i is conditionally dependent on all the variables in $(\mathrm{X_1},\ldots,\mathrm{X_{i-1}})$. There are many other rules for probability, such as the ***expansion rule***, named ***marginal probability***, considering the situation where X and e are random variables with k as a possible outcome:

$$p(X) = p(X|e^{k=1}).\,p(e^{k=1}) + p(X|e^{k=2}).\,p(e^{k=2}) + \cdots + p(X|e^{k=k}).\,p(e^{k=n})$$

$$= \sum_{k=1}^{n} p(X|e).\,p(e) \;\ldots\ldots (3.6)$$

This rule works to offer only the probability of one variable (X); all the information about the other variables (e) is ignored and this is not suitable for inferring our pre-crash system as reasoning under uncertainty or as a causal system, but it is used to find the probability of one variable only.

And there is the ***Independence rule***, where two random variables X and Y are independent and written as$X \perp Y$, if and only if:

$$p(X,Y) = p(X).\,p(Y) \;\ldots\ldots (3.7)$$

3.5.2 Dynamic Bayesian Networks

Dynamic Bayesian Network (DBN) is useful for problem domains where the state of the world is dynamic. The static BBN used to solve static problems only. Which means; every variable has a single and fixed CPT. This proposition of the static world is not always suitable and sufficient, some problems in the world need to be more flexible, do not have fixed variables or changeable values over time. DBN is a BBN, that is extended with a time dimension and which can be used to model dynamic systems [112, 114, 115].

3.5.2.1 DBN Graph

A DBN is a directed acyclic graphical (DAG) model, but it is based on a stochastic process [116], which consists of a time-slices concept, with each time-slice containing its own variables.

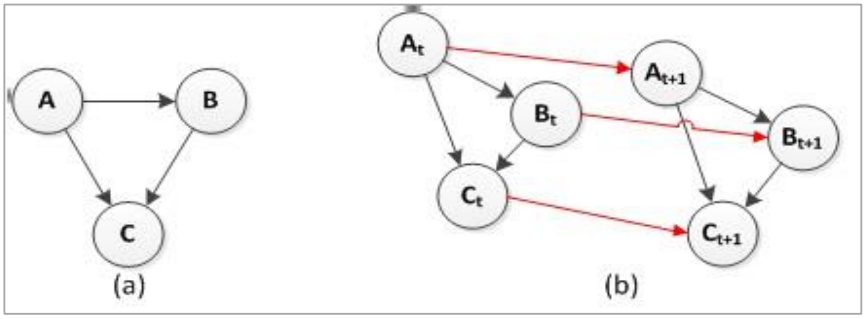

Figure 3.1 (a) initial network for the static BBN. (b) The two slice temporal Bayesian network for DBN.

A DBN is described as the pair (B_t, $B\rightarrow$), where B_t is a BBN that defines the prior or initial state distribution of the state variables $p(D_t)$ [112]. Typically, D_t= (C_t, A_t, B_t) represents all the variables of the system (input, hidden and output). $B\rightarrow$ is a two slice temporal Bayesian network that defines the transition model $p(D_{t+1}|D_t)$ as define in Equation (3.8):

$$p(D_{t+1}|D_t) = \prod_{i=1}^{n} p\left(D_{t+1}^{i}\middle|pa\left(D_{t+1}^{i}\right)\right) \ldots\ldots (3.8)$$

Where:

- D_t^i is the i-th node at time t, and could be a factor of A_t, B_t or C_t.
- $Pa(D^i_{t+1})$ are the parents of D^i_{t+1}, which can be in the same or the next time-slice.
- n number of nodes.

There are no parameters associated with the nodes in the first slice of the two slice temporal Bayesian network. The nodes in the second slice do have a CPT the structure repeats and the process is stationary, so the parameters for slices t = 2, 3 . . . remain the same. This means that the system can only fully described by giving the first two slices. In this way, an unbounded sequence length can modelled using a finite number of parameters. The joint probability distribution for a sequence of length T can obtained by *unrolling* the two slice temporal Bayesian network:

$$p(D_{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{N} p\left(D_{t+1}^{i}\middle|pa\left(D_{t+1}^{i}\right)\right) \ldots\ldots (3.9)$$

The (B_t, $B\rightarrow$) definition for this DBN is shown in Figure 3.1.

3.5.2.2 Conditional Probability Table of DBN

Each node in the DBN has a CPT that defines the Conditional Probability Distributions (CPD) of the represented discrete random variable. In DBN, the CPT of any variable in the network is the same as in a static BBN at the first time slice. However, the main difference is represented in the other time slices (i.e. $t+h$), while the CPT is built of the joint probability of conditional probability distributions, allowing it to create a CPT with the state of the parent or the variables in the previous time slices. This allows a decision about the new value after changes occur (i.e. new evidence is detected) over time. This will discussed in details in the implementation section of chapter 5.

The following two methods can be used to obtain the probabilities of the states for each node in the network [57, 111, 117, 118]:

- Obtaining the values by perform statistical analysis of a huge amount of training data. Training data to obtain by performing several tests in a test-bed specifically designed for the system and collecting the output for each test.
- Parameterising the network can be done using syntactical data from several published papers that are related or similar to the system, crash reports and transportation standards.

It was too difficult to acquire a large amount of training data for this study, as no test-bed is equipped with all the necessary sensors for the system; nor does any previous study contain the data required to parameterise the system. Therefore, the CPT values in the proposed system were obtained by gathering real high-fidelity data, as collected by the UK DfT in the form of crash report datasets [1, 2, 107] and a wide range of published papers and research [48, 57, 66, 68, 112, 119-127].

3.5.2.3 Inference (Reasoning with DBN)

To find the probability for a hypothesis node or variable depends on some evidence given in dynamic systems (over time), the inference has been performed on DBN; there are several inference methods that differ from static BBN because of the DBN property of working over time. The main goal of inference in DBN is to calculate $p(\boldsymbol{X}_t|\boldsymbol{Y}_{1:\tau})$. The most common types of inference are:

- ***Filtering over (t),*** calculation of $p(\boldsymbol{X}_t|\boldsymbol{Y}_{1:t})$ to achieve current beliefs dependent on all past evidence, where $\tau = t$. Filtering is used to track any current state to make normal decisions.
- ***Prediction over (t+h),*** calculation of $p(\boldsymbol{X}_{t+h}|\boldsymbol{Y}_{1:t})$ to achieve a future belief state dependent on all past evidence, where $h > 0$ and $\tau < t$. This method of inference can be used to evaluate the effect of possible actions on future time.
- ***Smoothing over (t-l),*** calculation of $p(\boldsymbol{X}_{t-l}|\boldsymbol{Y}_{1:t})$ to achieve a belief state in the past dependent on all evidence prior to the present. This method of inference is used over fixed time-lag $l > 0$. It is a useful way to get a best estimate of past state, because more evidence is available at time t than at times $t - l$ and $\tau > t$.
- ***Viterbi decoding***, calculation of $arg \max_{1:t} p(x_{1:t}|y_{1:t})$ to achieve the most likely sequence of hidden states dependent on a sequence of given observations. This method is a different kind of inference from those outlined above.

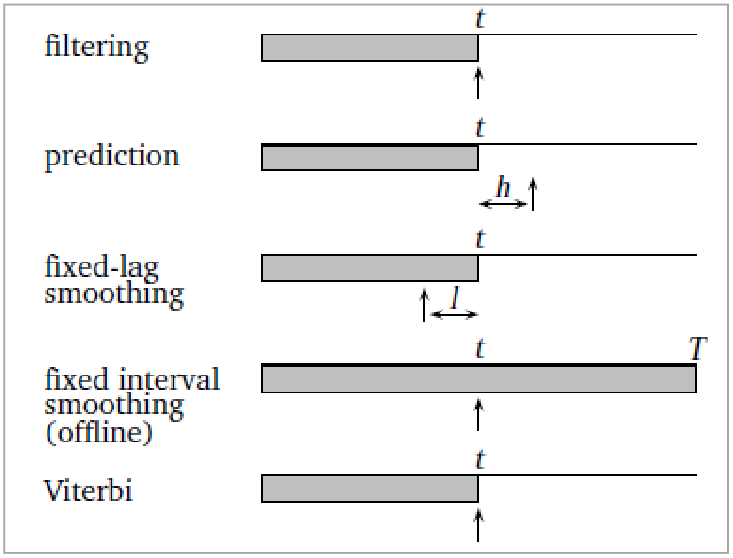

Figure 3.2 The main types of inference for DBN.

In Figure 3.2, which represents the main types of DBN inference, the region that is grey in colour is the interval containing evidence. The arrow represents the time of inference. It is current time, and T is the sequence length. These types of inference can work either with an Exact or Approximate inference;

Exact inference, this type of inference is simple and works to infer the DBN, but it first needs to unroll the DBN into a static BBN, before inference on the static BBN to find the values of the hypothesis nodes using clustering exact inference and a junction tree algorithm. This method takes advantage of the DBN structure, by creating a junction tree for each time slice and performing updates for time slices, up to, and including the current time-slice using inter-tree message passing. Figure 3.3 illustrate an example of how to unroll DBN into a static BBN.

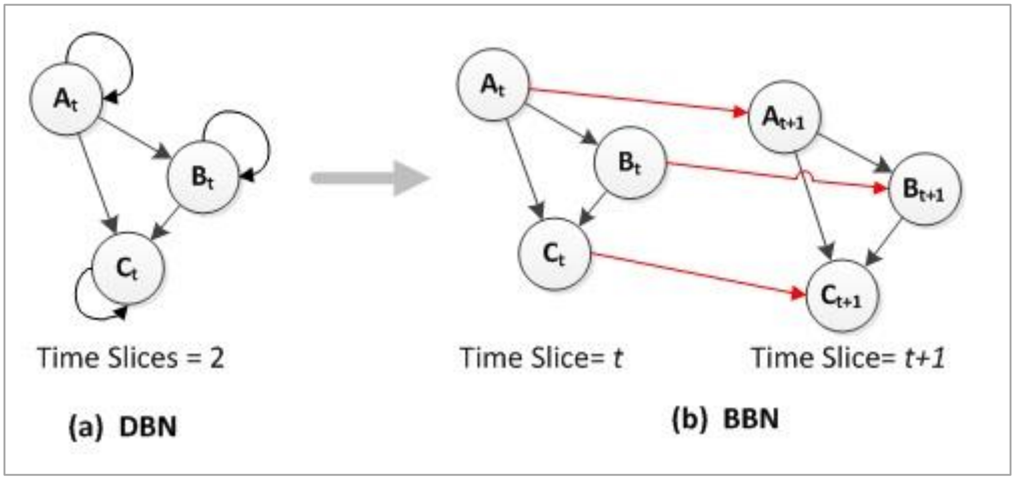

Figure 3.3 (a) Represents DBN. (b) BBN after Unroll DBN

- The ***Approximate inference***, this type of inference tries to unroll DBN to BBN then to convert it to Hidden Markov Model (HMM) and then apply a forward-backward algorithm. This is a useful method because it is implemented when the values of the BBN nodes are not available and need to be filled in, in an approximate way using algorithms for the missing values (e.g. Noise-OR algorithm) with larger networks. Approximate inference has more than one algorithm that it can deploy to find the BBN inference, and it depends on BBN considerations to select a suitable one (i.e. stochastic simulation algorithm, model simplification methods, search- based methods and loop belief propagation).

The proposed system performs inference to predict a crash in future time, so that the prediction over *(t+h)* inference method is used to calculate $p(\boldsymbol{X}_{t+h}|\boldsymbol{Y}_{1:t})$ to achieve a future belief state, dependent on all the given evidence from the past (*t-1*) and the current time (*t*), to evaluate the effect of possible actions on future time (*t+1*) using the Clustering exact inference algorithm to obtain the values of the hypothesis nodes. It is necessary to unroll the DBN and acquire its BBN to deploy a Junction tree algorithm and infer the network, this will be discussed in detail in chapter five of this thesis.

3.5.2.4 DBN Learning

DBN has a very useful and important property, which is the ability to learn from observations and/or statistical data. Bayesian learning calculates the probability of any hypothesis' node from given data (observable events). The probability of each hypothesis h_i, given data d, is given by:

$$p(X_i|d) = \alpha \, . \, p(d|X_i) . \, p(X_i) \, \dots \dots (3.10)$$

Where:

α denoted the unknown variable,

d denoted the given data, and

X_i denoted the hypothetical variable.

There are two types of learning DBN; ***Parameter*** learning, and ***Structure learning***. In parameter learning, the DBN structure is fixed and given by a system designer, but only the parameters are learned and updated in the CPT's values, although, when the structure is learned the network can be amended to suit a new structure.

Parameter learning: A DBN contains discrete random variables, and each variable has a CPT. All CPT values are unknown at until they are learned. The problem with this approach is that it needs a large data set to get satisfactory results; this data set, however, is not usually available.

Structure learning: The structure of the DBN can also be learned from a data set. The network structure can be derived from expert knowledge, after which a structure learning technique can be used to optimise the derived network structure [128, 129]. Structure learning has not been used in the proposed DBN and falls outside this thesis' scope.

3.5.2.5 Steps to creating a DBN

The following steps describe how to create a DBN:

1. Choose defining network nodes and their states;
2. Draw the causal relationships between random variables;
3. Determine the conditional probabilities for each node; and
4. Perform the inference process in order to infer the hypothesis node.

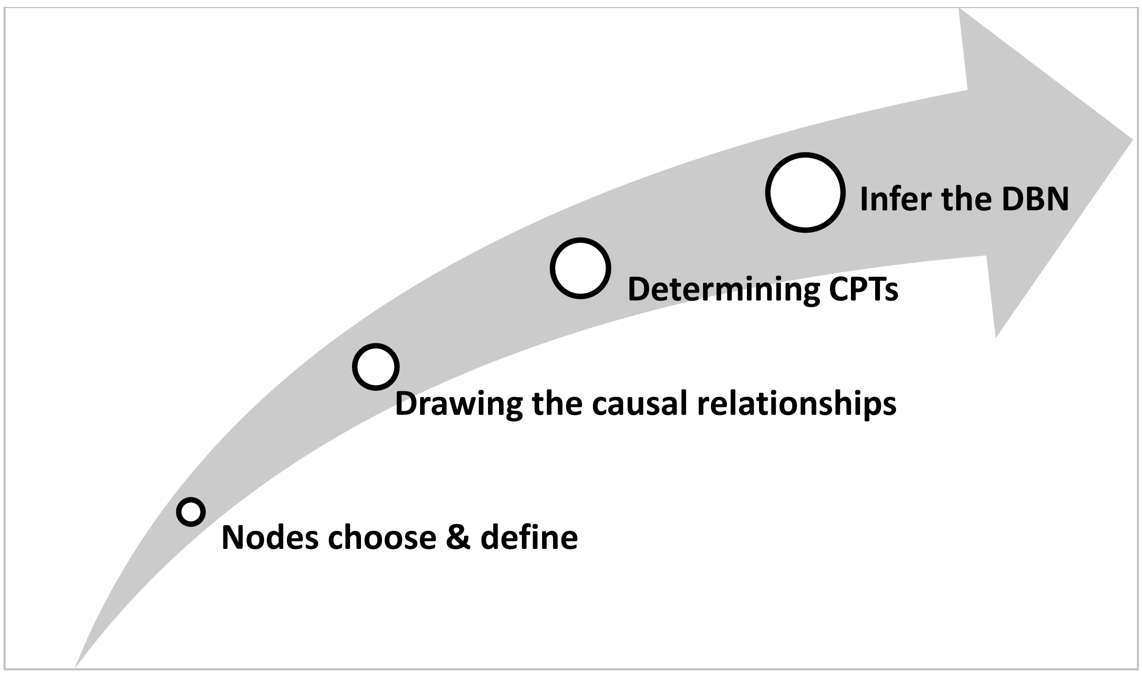

Figure 3.4 The four steps to create a Dynamic Bayesian Network

These steps are used to create the DBN for the system proposed in this thesis, as discussed in full detail in chapter 5.

3.6 DBN's Software Packages

In this section, an overview of the software developed at the Decision Systems Laboratory (DSL) has presented. Software that supports BBN and DBN functionality and properties (i.e. structural modelling, inference and learning, etc.) can divided into two kinds: software with a Graphical User Interface (GUI) and software without a GUI.

Software that does have a GUI is the most common and commercial and is reasonably easy for average users to use. Software without a GUI is generally academic-oriented; it is very flexible, but this can also make its application to specific problems very time-consuming and complex for a designer and coder. Here are some examples of the better-known BBN software packages:

3.6.1 DBN Libraries

Three libraries have a DBN functionality. All these libraries have a work-in-progress status, (e.g. many functionalities are missing and/or need to be tested and optimised).

3.6.1.1 The Bayes net toolbox for Matlab

Although Matlab software introduced the Bayes net toolbox (BNT) [130], it lacks usefulness because it does not support a free general purpose software library to handle different variants of graphical models, inference and learning techniques. The BNT can be used to build free, open source, and easy to extend libraries for research purposes. The library implementation can easily handle Gaussian random variables. The main Matlab disadvantage is that, it is awfully slow and without GUI. However, it has many advantages, such as;

- BBNs are represented as a structure with a graph, CPDs and a few other pieces of information.
- DBNs can be presented with a prior and an intermediary network, making it possible to only model first-order processes and offer some inference algorithms.

- The conditional probabilities of the selected variables can be continuous or discrete.
- The property of DBN learning for parameter and structure are also supported.
- The code is relatively easy to extend and well documented and the library has the best functionality of all the software currently available, which means it is widely in use.

However, greater functionality still needs to be added to make it a genuinely general purpose tool. Supplementary to this, the BNT is released as open source software.

3.6.1.2 The Graphical Models ToolKit (GMTK)

GMTK has become specialised for developing DBN-based automatic speech recognition (ASR) systems [131]. GMTK is a freely available and open source toolkit written in c++. It has some features that can be used to model large sets of statistical models (i.e. several inference techniques, continuous and discrete observation variables). The DBN is specified as a template that contains a number of time slices, specifying a special purpose language. Over time the time slices were unrolled to create an unrolled DBN. The GMTK is a hopeful library, but it is not a general purpose toolkit, because it only specialises in ASR, as it is lacks serviceable applications. The GMTK documentation is not complete and it is hard to find details about it [132].

3.6.2 DBN Modelling Tools

There are only two modelling tools that support temporal reasoning and have a GUI: these tools are; *BayesiaLAB* and *Netica.*

3.6.2.1 BayesiaLab

BayesiaLab is the most famous modelling tool, and consist of two modes: The *modelling mode* used to design the BBN, with the capability to add temporal arcs to indicate the parent node over time. This mode can design first-order Markov models only. While, the *validation mode* is used for inference. There are many methods of validation that should be selected to track changes to a system over time. The designer has to specify the initial state and temporal probabilities of each node in the BBN. In this toolkit, only the values of the parameters are changed without any graphic alteration to the DBN. This toolkit also has the facility to set evidence by hand or by importing a data file. The BayesiaLab software toolkit is developed by the French company Bayesia [133].

3.6.2.2 Netica

Netica is used to design static and dynamic BBNs; after the design is complete inference is possible. In a DBN there are temporal arcs that can be added between the nodes. Netica can explicitly define temporal relationships (time-delay) for two nodes in one node. In this way, a *k*-order Markov processes can be modelled when the DBN definition contains loops and the unrolled version should not contain loops. This can be solved by unrolling it for *t* slices and compiling its inference. Netica supports all inference methods, and can enter evidence by hand or by importing data files. Netica is a product of Norsys [134], which is located in Canada.

3.6.3 Decision Systems Laboratory software (DSL)

The decision Systems Laboratory is a research group within the Department of Information Science and Telecommunications and is an Intelligent Systems Program at the University of Pittsburgh. SMILE and GeNIe are members of the DSL family. They were developed for the purpose of probabilistic, structural Modelling, Inference, and Learning Engines. The software, the documentation, and many example applications can be freely downloaded [135].

3.6.3.1 GeNIe

GeNIe is the graphical user interface for the SMILE library. It is also implemented using c++ and Microsoft Foundation Classes. Its importance lies in the fact that it is easy to access and has a user friendly interface, making decision theoretic models axiomatic using a graphical interface approach (via click or drag and drop). GeNIe version 2.0 is a multipurpose application with a friendly environment for designing and building graphical decision models and for performing predictions and judgments using DBN and clustering inference algorithms for single and multiple BBNs.

3.6.4 Software Package Selection

After looking in depth at existing software packages, as shown in section 3.6. as a conclusion, there has been no comprehensive solution for modelling using DBNs until now. In this thesis the GeNIe v2.0 has been used; it is the most suitable software for implementing a pre-crash DBN with the specification that are required, as will be shown in chapter 5.

Some packages have common features such as unmanageable, slow and restricted scripting and implementing, traditional definitions, absent importing and exporting functionality, implementations for complex models (i.e. GMTK, BayesiaLab and Netica). Meanwhile, Bayes Net Toolbox (BNT) applies the most packages, which then support BBN and DBN functionality. However the implementation is slow, as they do not have a GUI, and rely on a very traditional definition of BBNs and DBNs. However, BayesiaLab and Netica are not satisfactory either, because of a lack of functionality and they are also slow, making them useful for designing small BBNs but not complex models.

The main reasons for choosing GeNIe v2.0 are [135]:

- It supports DBN implementation by providing temporal reasoning so a DBN can be implemented.
- It provides a clustering inference algorithm, which is the algorithm selected to perform the inference in our system.
- It provides bar charts to assist the nodes to show the probabilities for each node state graphically.
- It is simple to design the DBN and to define temporal nodes using a graphical user interface.
- It can open multiple networks at the same time.
- It supports diagnosis.
- It offers text formatting support (font type, size and colour)
- It supports complete integration with Microsoft Office (Importing and exporting dataset).

3.7 Summary

This chapter presented the theoretical foundation of the reasoning using DBN. A global overview of Static BBN and DBN theories and techniques that is very important for understanding the reasoning informing the methodology and the selection of DBN as a reasoning algorithm. In this chapter, the Static BBN and DBN functions are presented in detail, such as network design, inference, and learning. Furthermore, a short justification of DBNs was given. As such, this chapter is useful for any reader whose aim to become familiar with static and DBN theory and techniques.

This chapter has effectively presented the existing software packages with the ability to design DBN and execute its functions and present the DBN properties. After which, an investigation and justification was introduced to select the best software available at present, when designing our DBN and performing the implementation of reasoning. As a conclusion to this investigation, SMILE and GeNIe are the best software in the field of reasoning. Meanwhile, the model design and implementation approach in GeNIe will be discussed in chapter 5.

Chapter 4 - Context Aware On Board Unit Architecture

Chapter Objectives:

1) Provide an overview of general context aware Pre-crash system architecture.
2) Define the components of the proposed pre-crash system architecture.
3) Describe the actions and services provided by the proposed Pre-crash system.
4) Present the mechanism of the Pre-crash system architecture.

4.1 Introduction

This chapter describes the designing of a context aware pre-crash system architecture in VANET, which is capable of adapting its operations to the current context without user interaction, it is thus aimed at augmenting usability and effectiveness, by taking into account contextual information drawn from the environment.

A general context-aware system architecture is designed for the pre-crash system, deploying a DBN algorithm for reasoning. A DBN is used, as shown in chapter 3, to enhance the reasoning process in the context-aware system, taking into account uncertainty in the contextual information detected, and learning model parameters using DBN learning property. The proposed system is suitable for any region or country in the world because its DBN likelihood can be updated using any dataset from any region to be more accurate and real.

This context-aware system architecture is a general architecture in VANET, which means it is usable for any reasoning mechanism (i.e. Fuzzy logic, Neural Network, etc.), in the proposed system the DBN algorithm has been deployed as will be discussed in chapter 5.

4.2 Overview of Context Aware On Board Unit Architecture:

In this section, a general context aware architecture is presented. As depicted in Figure 4.1, the context aware architecture incorporates three main phases: A physical phase, a thinking phase, and an application phase, all of which together represent the three main subsystems of a context aware layered conceptual framework; Sensing, Reasoning and Acting subsystems [136]. The presented OBU architecture is design based on the Layered Conceptual Framework architecture. The red callouts in Figure 5.1 represent the mapping of the proposed architecture to the Layered Conceptual Framework as shown in Figure 2.6 in Chapter 2.

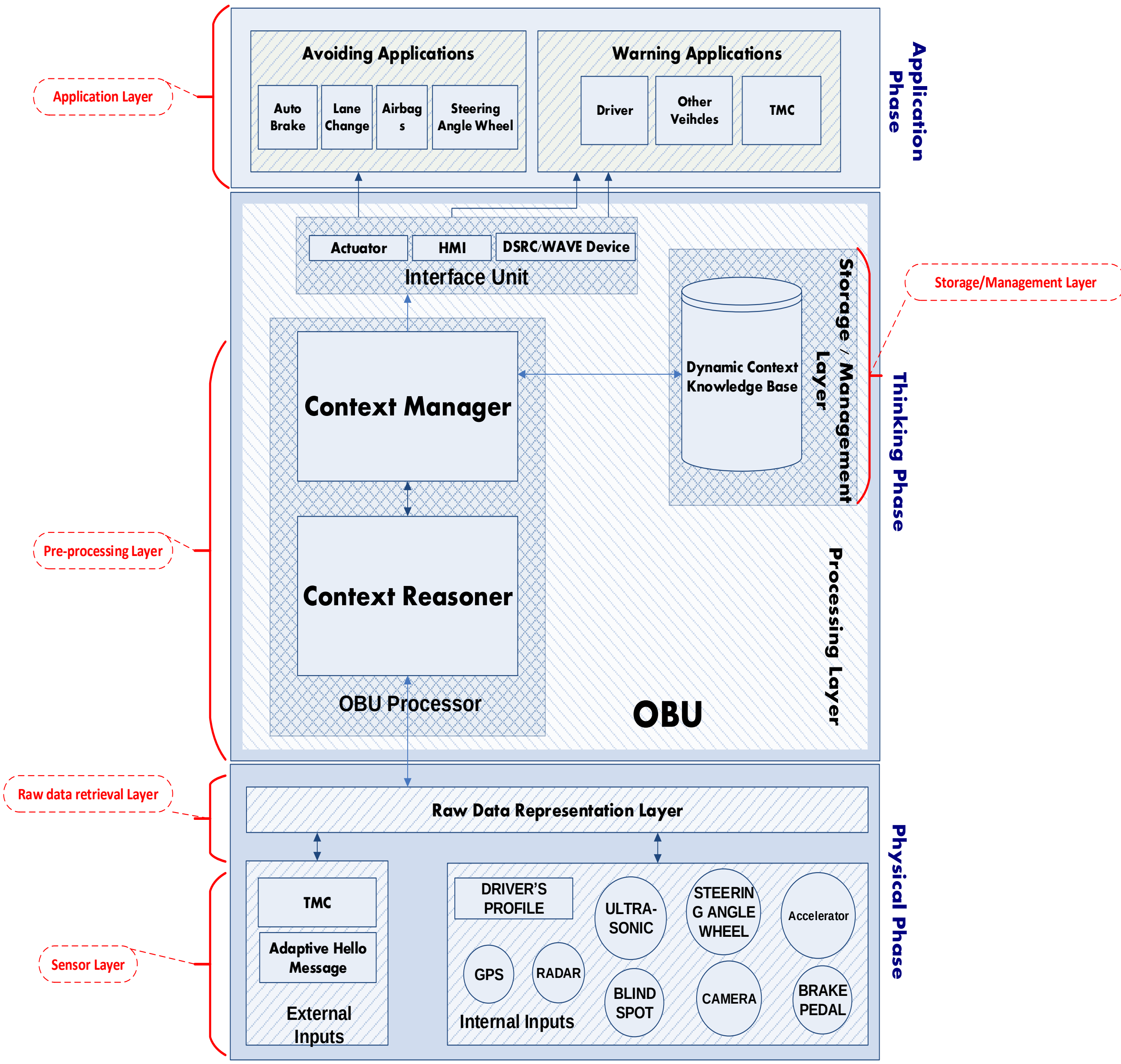

Figure 4.1 Context Aware Pre-Crash Architecture.

4.3.1 Physical Phase

The sensing phase is responsible for gathering low-level contextual information. This phase contains the following two layers: sensing and raw data layers.

4.3.1.1 The Sensing Layer

This layer is responsible for acquiring the context data using different types of sensors. It consists of two types of sensor: delivering internal and external inputs as shown in Figure 4.1. A detailed illustration of each sensor is depicted in the next section [3, 7, 20, 90, 91, 137-143].

i) Internal sensors:

These sensors are responsible for sensing all the data about the vehicle and the surrounding environment (i.e. speed, blind spot, distance, road surface, etc.), as shown in Figure 4.1. The internal sensors include the following:

- **Camera:** The camera sensor is responsible for capturing contexts about lane detection, lane deviation, lane departures, difficulty detection, road collision monitoring and driver distraction monitoring [5, 7].
- **GPS:** Global Positioning System (GPS) provides extremely accurate location data and time information. GPS helps to provide current location, direction, and angle of the vehicle [12, 137]. This factor is important to check if the vehicle's speed is in the, over or below the speed limit. Studies showed that crashes possibilities are affected highly with reducing vehicle speeds by a more than 5km/h, which would result in a 17.3% reduction in all serious crashes [123]. Road speed limit has good affect if vehicle speeds decreased by 5km/h on arterial roads with a speed limit of 60 km/h, there would be an 11.9% reduction in all serious crashes [1, 124].
- **Brake Pedal:** A brake pedal sensor provides the status of the brake pedal at a given time, and can help the vehicle’s hydraulics to engage to slow or stop the vehicle. In addition, two other functions can also occur, one of which is critical to the operation of the vehicle; i.e.

disengaging the cruise control. The other function is to switch on the brake lights to alert other vehicles that the car is slowing down and/or is going to stop [17].

- **Steering Angle Wheel:** The steering angle wheel sensor provides the value of the angle of the steering column of the vehicle. This information is useful for determining the vehicle’s direction, and it is considered as one of the most important factors for describing driver behaviour [12, 22].
- **Radar:** Radar is an object-detection system that uses long-range radio waves to determine the range, altitude, direction, or speed of an object. It can be used to detect vehicles travelling over a long distance [90, 136].
- **Ultrasonic:** Ultrasonic sensors are able to detect objects, such as vehicles and pedestrians, during low-light periods (night, dawn, dusk) and in the presence of obstacles and shadows [20, 137].
- **Blind spot**: Blind spot sensors are responsible for detecting other vehicles from the left and right sides to indicate whether it is safe to change lane or not. This can work in all weather conditions over short distances, and is still being developed and improved upon [20, 140].
- **Driver’s Profile:** This profile contains all the information about the driver such as; age, gender, weight, medical and behaviour history information of the driver [138]; this is very useful with safety systems, because it provides the system with contextual information that may affect decision making (i.e. driver’s age may be a good factor on which to base a decision) [141].

ii) External sensors

External input must also be considered to support the context, as captured from the internal inputs, so as to make the right decision about the risk. The input data are shown in Figure 4.2.

- ***TMC (Traffic Management Centre):*** TMC provides dynamic traffic information for motorists through a variety of sources, including variable message signs, the internet, WAP-enabled mobile phones, and local media. TMC can provide several information types; giving information about the outside environment, traffic information, weather status, etc. [17, 139]. TMC can also help to improve and evaluate the traffic situation and provide information to the travelling public.
- ***Adaptive Hello Message:*** using the HELLO message concept to gather information from other vehicles on the road (i.e. position, direction, speed) [142, 143].

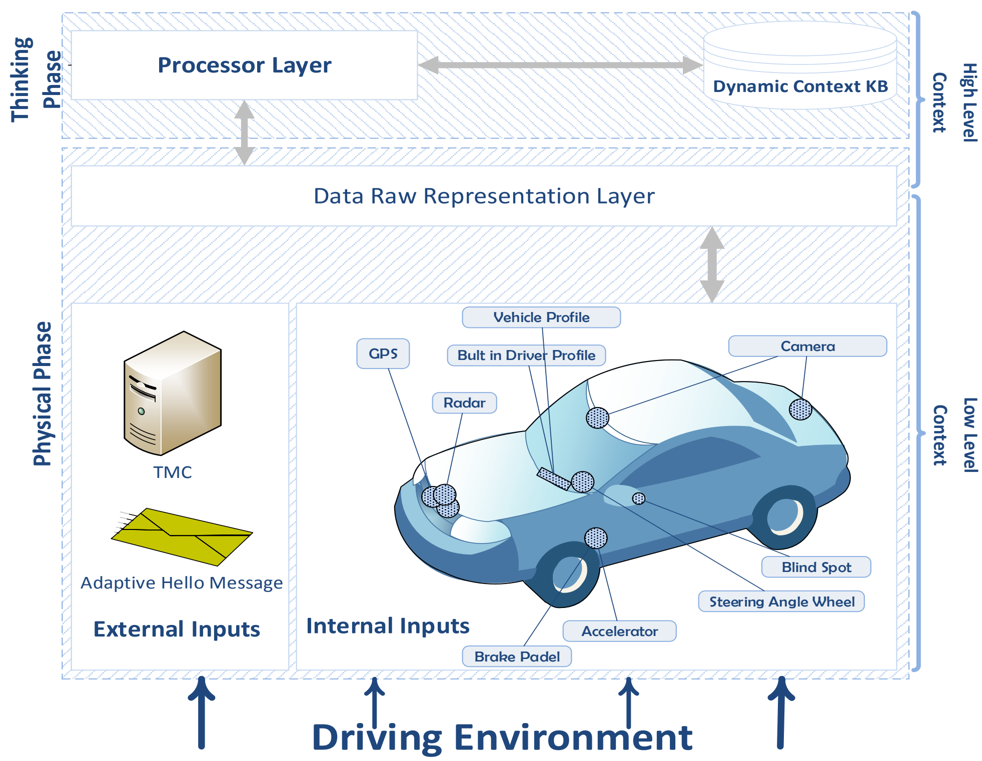

Figure 4.2 Pre-crash system architecture sensors

4.3.1.2 The Raw Data Layer

This layer is responsible for separating low level sensing details from sensors for the upper layer of the system, as well as for abstracting the contextual information received from the sensor layer. This layer is responsible for converting the sensed context into machine-readable form. This will make the context understandable and useable, to prepare the machine for analysis and process this context at the next layer (the thinking phase), as shown in Figure 4.2. This layer is also responsible for the re-collection and reusability of the contextual data requested by other applications; alternatively, it can provide new services to the user.

4.3.2 Thinking Phase

This phase comprises the management of contextual information. This can vary from a simple mechanism for querying the data to powerful reasoning, including the inference of deduced context. There are two types of contextual information: certain contextual information, which is obtain from a single sensor, and uncertain contextual information, which cannot acquire by a single sensor and which may be incomplete or inexact. The prediction of crash likelihood and severity is considered uncertain form, and as needing to be extracted from among contributory factors. The reasoning phase consists of two layers: processing and storage/management layers.

4.3.2.1 Processing Layer:

This layer is responsible for extracting information about hazardous situations. The processing layer consists of two sub-layers: context reasoning and context management.

i) Context Reasoner (CR)

This component is responsible for evaluating the exact situation and reason for uncertain contextual information in order to make decisions about it. The CR’s task is to predict the crash likelihood and severity levels as a first step to crash prevention; the design, implementation and verification for this stage are achieve by using a DBN. This is illustrate further in Chapter 5.

ii) Context Manager (CM)

This component is responsible for triggering an appropriate application in order to act upon the information being forwarded by the CR. This processor responsible for the second step ot crash prevention by recive the infer results about the crash likelihood and its severtiy level, then it sould select the perfect solution to either avoid the crash or mitigat the crash severity by deploying an algorithm to deploy the suitable application. The algorithm that must be used in this processor is out of this thesis scop and we consider it for the future work.

This processor is connected to the dynamic data base context using bidirectional arow to store, retrive and learn different type of data (e.g. historical: driver behaiver, vehicle service, road acceidents…etc.). and also, it cosider as the next stage after predicting the crash happening and severity to deploy the suitable action.

iii) Interface Unit (IU)

This unit receives the final decision from the context manager regarding which application should be used to avoid the crash. This unit contains:

- **Actuator:** The actuator is the device responsible for carrying out specified control of the vehicle without any command from the driver. A message, including parameters, should be send for the purpose of emergency assessment.
- **Human Machine Interface (HMI):** The HMI is responsible for issuing driver warning and selecting a suitable means of warning the driver; e.g. alarms, harsh noises, seat vibration or simple messages.
- **DSRC/WAVE**: DSRC/WAVE is a network device based on IEEE 802.11p. It is responsible for connecting the vehicle with other vehicles or a road side unit RSU [142].

4.3.2.2 Storage/Management Layer:

This unit is responsible for storing all the data and information needed to support the context manager layer. It forms a dynamic context knowledge base, which is extendable and regularly updated; i.e. it can be adjusted to update the stored information, including maps, driver behaviour and historical crash data [144].

This Dynamic Context Knowledge Base may consist of:

- **History Base:** this database contains the contextual information history about the driver, roads, collision patterns, etc. The History base is useful for assisting our system to improve its learning properties. It works with the Pre-crash algorithm (DBN) to update and learning the CPT’s values. The prediction property of our model could not be perform without the support of a historical foundation. For example, the application relies on the location history of the vehicle in order to predict its future location; thus, it must have historical data.

- **Digital Road Map:** this database contains maps, a routing table and representations of a particular area, all of which are stored in the Dynamic Context Knowledge Base. A digital road map lists the routes to a particular network destination, and the distances associated with those routes. The routing table contains information about the topology of the network. A digital road map is dynamic and can be update after a new route is discover on the network after collecting data from the GPS or TMC, which is then calculated in the processor to obtain updated information. The digital road map is manage by the Knowledge Manager.
- **Built-in Vehicle Profile:** This built-in profile contains information about the vehicle; this information are provided by the vehicle manufacturer and comprises tyre-road friction, time to stop, time to crash TTC, vehicle weight, vehicle size, ID, colour and distance for crash to avoid. Some information is provided by the data mining component in the thinking layer, such as history and behavioural information [144].

4.3.3 Application Phase

This phase represents the acting subsystem in a context-aware system. It is responsible for disseminating warning messages that include corrective actions for the other vehicles on the road. It also operates in-vehicle alarms to warn the driver to prevent the occurrence of accidents and to decrease the number of potential fatalities. Two types of applications were apply using the pre-crash system architecture, in accordance with the decision made by the context manager: avoidance and warning applications.

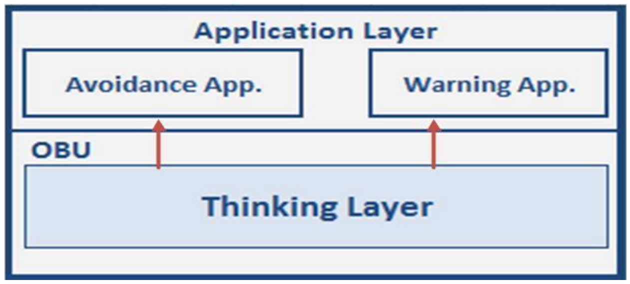

Figure 4.3 Application Layer Components

- **Avoiding applications** are responsible for taking action to avoid or mitigate an abnormal situation, such as a collision. This can be achieve by applying a number of applications, such as auto-brake, auto-stop and lane change.
- **Warning applications** are responsible for warning drivers via seat vibration, tightening seatbelts, on-board messaging or voice-based alarms. These applications can also warn other vehicles travelling on the same road or in the same direction about an abnormal situation using an adaptive HELLO message. Figure 4.4 depicts the mechanism for predicting crashes and for calculating corrective actions for other vehicles on the road.

4.3 Pre-crash system mechanism

This section depicts the mechanism for predicting pre-crashes and calculating corrective action for other vehicles on the road, also describing the data flow between architectural components, from the start point to the end point. As shown in Figure 4.4. Firstly, the system starts by gathering data from the surrounding environment via internal and external sensors; such as vehicle distance, speed limit, etc. The system then transmits this contextual information to the raw data representation layer to represent it in usable form. Then the Context Reasoner (CR) receives the contextual information to evaluate the exact situation.

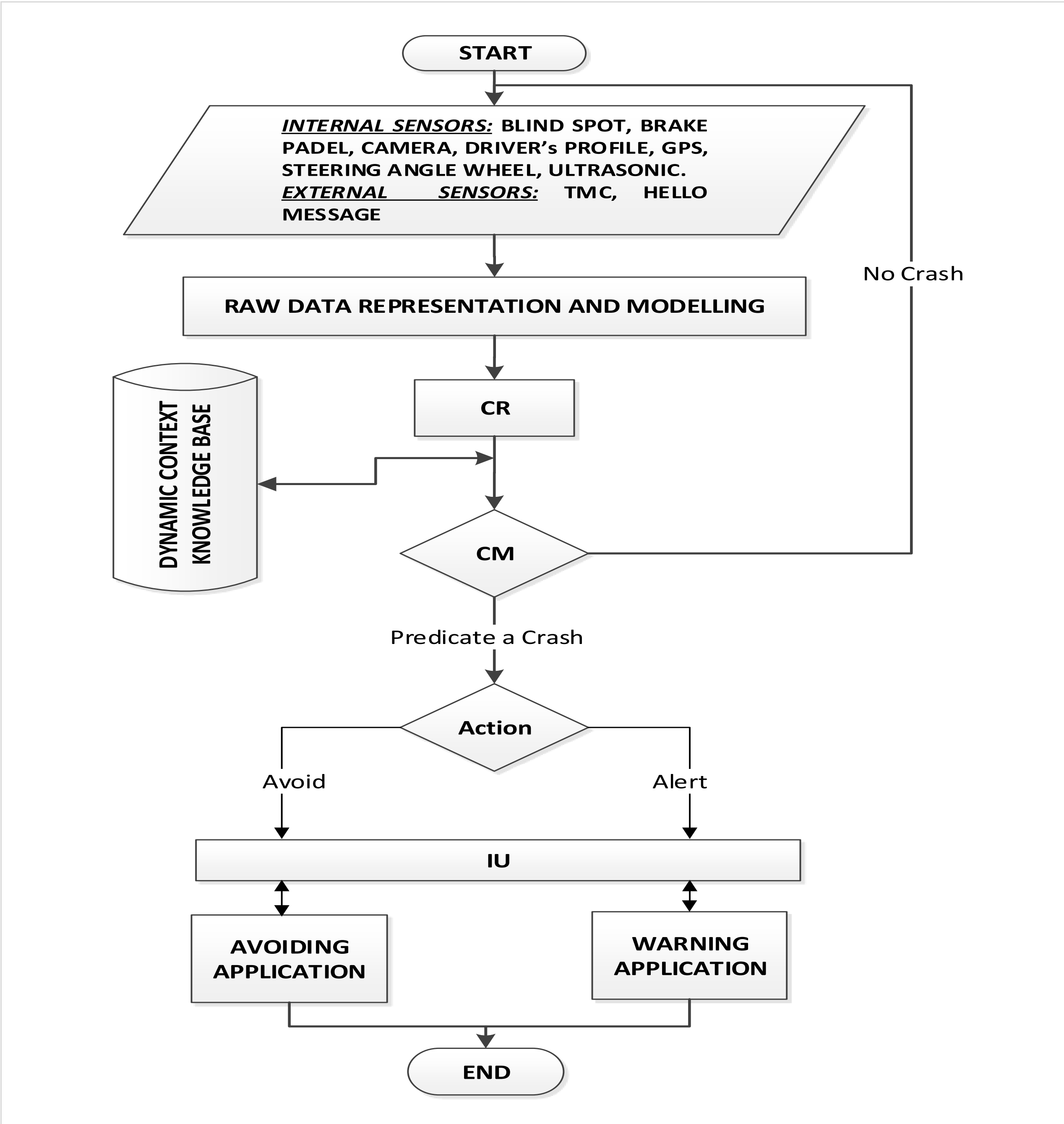

Figure 4.4 Pre-crash system mechanism

After the situation has been describe, the Context Manager (CM) determines the accuracy of the crash likelihood. Then the system has to select a more suitable action to avoid or mitigate the crash via the Interface Unit (IU), and pass this decision via the application phase to deploy the correct action (i.e. select a suitable application to run).

The CM relates to the Dynamic Context Knowledge Base and can be used to store and retrieve context information, which is required to store the driver's and vehicle's history, the dataset for the road and the traffic infrastructure for the region the vehicle is travelling in.

4.4 Summary

In this chapter, a novel context aware pre-crash system architecture has been propose. The proposed context-aware system architecture based on context-aware systems five layers conceptual framework. All the components of the architecture have been propose in detail, as a general architecture that can be integrate using a reasoning under uncertainty technique. In our system a DBN is deploy as a reasoning algorithm in the processor layer as a special reasoning algorithm; the details of this were present in the previous chapter (Chapter 3).

A system architecture mechanism is also presented to explain how the contexts sensed are used and converted from low level to high level contextual information, and to explain the relationship between architectural components from the lowest level (physical phase) to the highest level (Application phase).

Chapter 5 - Pre-crash System Designing and Developing

Chapter Objectives:

1. Determine the objectives behind the development of the proposed DBN model.
2. Propose the novel DBN model for crash prediction.
3. Define the necessary steps for the creation of the proposed DBN model.

5.1 Introductions

This chapter presents a novel probabilistic framework for predicting the crash likelihood and severity level in VANET, using a Dynamic Bayesian Networks (DBN) to fuse contextual information about the driver, vehicle and environment, and then inferring the crash likelihood. The proposed system is able to predict vehicle crashes and their severity; fatal, serious or slight crash, depending on different contributory factors. These factors cannot be considered to be unambiguous input into the system, as high-level contextual information cannot be inferred using normal sensors. Thus, it is essential to capture these temporal low-context factors and to incorporate the evidence over time. Consequently, the accurate and effective prediction of different types of crash requires different types of context to be sense and gather. In fact, the inaccurate and incomplete sensed context information needs to be fuse with other factors in order to make it more accurate. There are a number of methods of information fusion: fuzzy logic, neural networks and the Kalman filter. None of these methods efficiently support the sensed incomplete data, uncertainties, variables dependencies or the temporal feature demonstrated by the proposed system [66].

The proposed system in this thesis employs DBN to gather data from different types of sensors in order to predict crash likelihood at different levels of severity [145, 146], because DBN consider comprising the most reliable method for managing inaccurate and unobservable inputs, and due to their ability to work over time [68, 147]. In addition, DBNs are efficient at using more than one context and combining them to realise high-level contextual information in order to infer reason about uncertain contexts [66, 145], as shown in Chapter 3.

5.2 Problem definition

The main objective of designing the DBN model is to predict the crash likelihood and severity level by inferring the unobserved high-level context from the observed context. Vehicle crashes are affected by various contributory factors; the vehicle, driver and environment. It is impossible to cover all these contributory factors in this system. However, this thesis deals with the most important factors which positively affect the possibility of a crash and may lead to more accurate prediction of crashes than others; these factors are information and observable factors, as will be shown in Section 5.3.1. Therefore, the method that has the best capability of working under the continually changing conditions is the DBN, which enables each state to be inferred by sensing and combining a large amount of context. The crash is therefore considered as the current state.

In order to predict crashes, a DBN was employed to model the probability distribution over the information and observable variables, so as to infer the crash state of the vehicle at a future time $(t+1)$. In this thesis, the unrolled DBN is considered, and the hypothesis node "Crash" node at time slice $(t+1)$ depends on the past crash state at time $(t-1)$, as well as on the current time state at (t) and the random variables at time (t) only, as in the following equation [112]:

$$p(C_t|\, C_{t-1}) = \prod_{i=1}^{n} p\left(C_t^i \middle| pa(C_t^i)\right) \ldots \ldots (5.1)$$

Where N is the number of nodes in the network, C_t^i is the i-th node at time slice (t) and $pa(C_t^i)$ are parents of C_t^i.

$$p(C_{t+1}|C_t) = \prod_{i=1}^{n} p\left(C_{t+1}^i \middle| pa(C_{t+1}^i)\right) \ldots \ldots (5.2)$$

Where N is the number of nodes in the network, C_{t+1}^i is the i-th node at time slice $(t+1)$ and $pa(C_{t+1}^i)$ are parents of C_{t+1}^i.

Equation 5.1 is used to find the infer of the crash node in the current time slices depends on the past time slice and Equation 5.2 is used to infer the crash node for the future time slice depends on the state of the current time slice. It is therefore important to apply a sequence of steps when designing a DBN model; the following section defines these steps in detail.

5.3 DBN-based pre-crash system

This section outlines the steps taken for the creation of the DBN pre-crash system, which is able to capture the temporal aspects of the crash and integrate the evidence over time. As mentioned in Chapter 3, four stages have to be carry out, in order to design a DBN, starting with selecting the network's nodes and ending with inferring the network to find the hypothesis node likelihood. A step-by-step explanation is presented in the next section, as shown in Figure 3.4.

5.3.1 Choosing and defining the DBN nodes

The hypothesis node in this network is the "crash" node, which includes four mutually exclusive states: fatal, serious, slight and no crash. The rest of the nodes are divide into two groups: information factors and observable factors. The first group (the information nodes) represents the variables, which may affect the crash node; and the second group (the observable nodes) corresponds to the information that results from the crash node. A brief description on each group and node is given in the following section:

Group 1: This group includes the contextual variables i.e. the information regarding the surrounding environment, road and driver, which affect the state node, and this information does not change suddenly over one time slice. These information nodes include:

1. AGE: Studies showed that driver's age affects crash possibility, which increases at ages that are between (66-75), and (19-35) [48, 119, 120, 147].

2. Gender: Studies showed that driver’s gender affects crash possibility, which increases according the driver's gender (male/female) for some specific age ranges [48, 146, 147].
3. Driver: The driver's node has been added, which categorises the driver as good or bad depending on their age and gender, to reduce the complexity of the proposed system.
4. DAY_OF_WEEK: Studies proved that the crash occurrence frequency gets higher on particular weekdays more than on weekend days [1, 146, 148].
5. TIME: Studies proved that crashes are more likely to happen at peak times than at other day times [1, 2, 121].
6. TRAFFIC_STATUS: This node has been added which represents the traffic status being either dense or normal which basically depends on the day and time.
7. WEATHER_CONDITIONS: Many studies proved that the crash is more likely to happen when the weather conditions are bad [1, 2, 122].
8. ROAD_SURFACE_CONDITIONS: Studies proved that the crash is more likely to happen when the road surface conditions are bad [1, 2, 121].
9. LIGHT_CONDITIONS: Studies proved that the crash is more likely to happen when the light conditions are bad [1, 2, 122].
10. ENVIRONMENT: this node has been added which represents the environment being either good or bad depending on the weather, road surface, light and traffic status, to reduce the complexity of the proposed system.
11. Road Type: this node represents the road type in UK according to the DfT report [1].

	No.	Node Name	No. of states	Node States
Group 1	1	AGE	5	- UNDER_18 - BTW_19_35 - BTW_36_65 - BTW_66_75 - OVER_75
	2	DAY_OF_WEEK	7	- SUN, MON, TUE, WED, THU, FRI, SAT
	3	DRIVER	2	- GOOD - BAD
	4	ENVIRONMENT	2	- GOOD - BAD
	5	GENDER	2	- Male - Female
	6	LIGHT_CONDITIONS	2	- Daylight - Darkness
	7	ROAD_SURFACE_CONDITIONS	5	- Dry, - Flood_over_3cm_deep, - Frost_or_ice, - Snow, - Wet_or_damp
	8	ROAD_TYPE	6	- Dual_carriageway, - One_way_street, - One_way_street_Slip_road, - Roundabout, - Single_carriageway, - Slip_road
	9	TIME	2	- PEAK_TIME - OUT_PEAK_TIME
	10	TRAFFIC_STATUS	2	- DENSITY - NORMAL
	11	WEATHER_CONDITIONS	7	- Fine_AND_high_winds - Fine_NO_high_winds - Fog_or_mist - Raining_AND_high_winds - Raining_NO_high_winds - Snowing_AND_high_winds - Snowing_NO_high_winds

Table 5.1 Pre-crash Dynamic Bayesian Network information nodes and their states.

Group 2: This group denotes the contextual variables that result from the state node. It includes observable factors, which comprise:

1. Brake_Pedal: This factor is very important because it significantly affects many factors, such as the vehicle's speed and the distance between the modelled vehicle and others. This factor has a direct effect on the distance and speed factors, as shown in Figure 5.1 [1, 125, 126]. Hence, this node has been selected as an observable node in the proposed system.
2. Distance: This factor has a very large effect on the crash probability because it represents the real distance between the modelled vehicles [1, 124, 126, 149]. The system will not perform the inference process if there is no frontal vehicle or the distance between the vehicle and the ahead one is more than 20 meters
3. Speed: This factor has a major effect on detecting the crash severity, depending on the vehicle's speed compared to the road speed limit; when the vehicle reaches the critical distance and is not stopping, a crash will surely occur [1, 124, 126]. (For instance, the risk of a casualty crash approximately doubles with each 5km/h increase in speed on a 60km/h speed-limited road, or with each 10km/h increase in speed on 110km/h roads) [150].
4. Lane: Studies have shown that a high percentage of crashes occur when both vehicles are in the same lane; when the lane state is different the crash possibility is very low and may reach zero [2, 127, 151].
5. Steering_Angle_Change: Studies have proved that this factor has a strong effect on the lane factor, because any steering change will lead to a change in the lane state [2, 127, 152].
6. Blind_Spot: Studies have proved that this factor has a strong effect on the lane and steering change states, because if there is any vehicle in the blind spot range, this will lead

to steering change, and if the steering angle changes the possibility of a crash is increased [2, 153-156].

	No.	Node Name	No. of states	Node States
Group 2	1	BLIND_SPOT	2	- ON - OFF
	2	BRAKE_PADAL	2	- ON - OFF
	3	DISTANCE	3	- less20m - less5m - less1m
	4	LANE	2	- Same_Lane - Different_Lane
	5	SPEED (road speed limitaion)	4	- OVER_LIMIT - LIMIT - BELOW_LIMIT - ZERO (no movement)
	6	STEERING_ANGLE_CHANGE	2	- YES - NO

Table 5.2 Pre-crash Dynamic Bayesian Network observable nodes and their states.

	No.	Node Name	No. of states	Node States
Hypothesis	1	Crash	4	- Fatal - Serious - Slight - No_Crash

Table 5.3 Pre-crash Dynamic Bayesian Network hypothesis node and its state.

5.3.2 Drawing the causal relationships and DBN graph

In this step, the conditional independence between the DBN nodes is decided after the variables have been chosen; this step includes the drawing of a directed acyclic graph. Figure 5.1 depicts the DBN structure at time $(t-1), (t)$ and $(t+1)$, and the conditional independence between the variables. The hypothesis node at time slice $(t+1)$ is affected by the information variables at time slice (t) and the hypothesis node at time slice$(t-1)$.

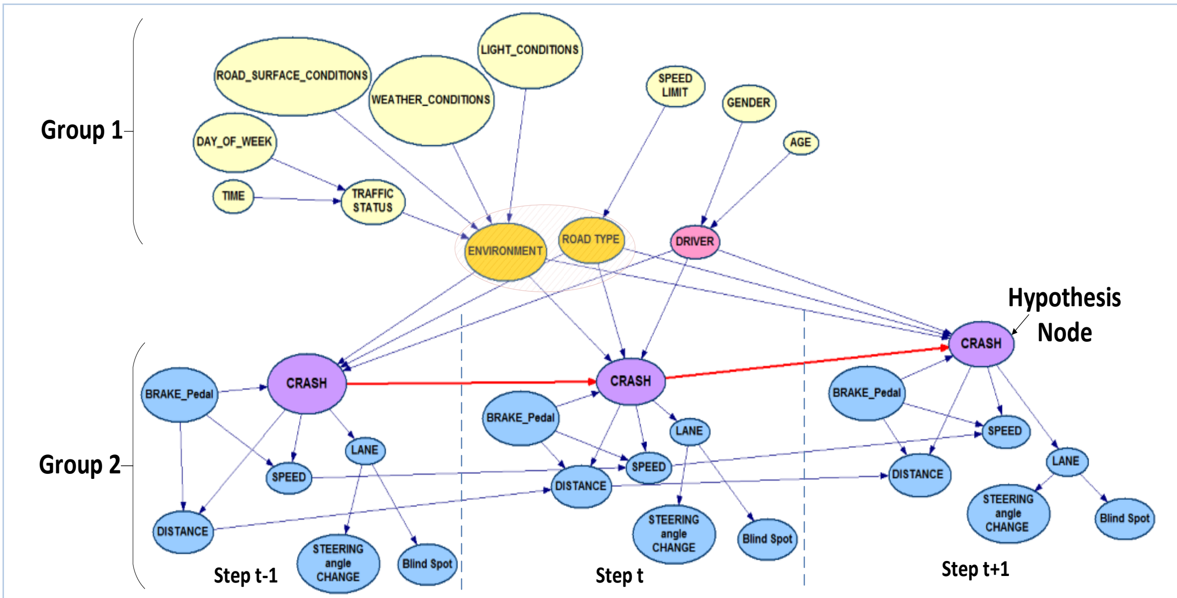

Figure 5.1 Dynamic Bayesian Network with dependencies between different time slices

As shown in Figure 5.1, group 1 represent the information factors, the hypothesis node, which is the crash node, represents the unobservable node, and the group 2 represent the observable nodes. In order to predict the future context of a vehicle, the use of a DBN was chosen. This network consists of different time slices which all contain an identical Bayesian network. The nodes between the time slices are connected with arrows to represent dependencies among these time slices. Figure 5.1 shows the DBN after unrolling it to be a static Bayesian network of three time slices, $(t-1)$, (t) and $(t+1)$, in order to infer it using the Bayesian Belief Network inference algorithm.

Tables 5.1 to 5.3 shows the DBN nodes' states. In the real world, there are very many factors that affect the likelihood of a crash occurrence, and it is impossible to cover all these factors in this system, as it would result in a very complex system. However, this thesis deals with the most important factors which positively affect crash possibility.

5.3.3 Parameterising the DBN

Each node in BBN has it is CPT that defines the Conditional Probability Distributions (CPD) of the represented discrete random variable. The entries in the CPTs tell us what the probabilities are of a hidden node given its parents. A potential problem of using CPTs for defining the conditional probability distributions is their size. The size of the CPT of any variable *X* depends on the number of states r_x of the variable and the number of states r_j of each of its parent, as follows:

$$size(CPT)_X = r_X \prod_{r_j = r(\boldsymbol{Pa}(\mathrm{X}))} r_j \ldots \ldots (5.3)$$

As can be seen from Equation (5.3), the size of the CPTs grows exponentially with the number of parents. By knowing this, it is important to keep the number of dependent nodes as low as possible. By using equation (5.3) the proposed network equation will be:

$$size(CPT)_{Crash} = r_{Crash} \prod_{r_j = r(\boldsymbol{Pa}(\mathrm{Crash}))} r_j \ldots \ldots (5.4)$$

The CPT size of our model hypothesis node (*Crash node*) depends on the number of states r_{crash} of the variable *Crash* and the number of states r_j of each parent which are denoted by *Pa(Crash).* If r_j is equal for all parents of *Crash*, the equation simplifies to:

$$size(CPT)_{Crash} = \mathrm{r_{Crash}}.(r_j)^{\mathrm{n}} \ldots \ldots (5.5)$$

Where *n* is the number of parents, the *Crash* node has three parents' nodes. Hence,

$$size(CPT)_{\mathrm{Crash}} = 4 * 6 * 2 * 2 * 2 = 192 \text{ Values, this is just for crash node CPT}$$

After choosing the DBN nodes, designing its graph, and specifying relations between the nodes the probabilities of CPT values with the prior probability of the root nodes and the conditional probabilities have to specify for each node in the DBN. Deciding the value for each node is main key for the network to infer the network and obtain the likelihood crash values. The following two methods may be used to obtain the probabilities of the states for each node in the network:

- Obtaining the values by performing statistical analysis of a huge amount of training data. Training data are obtained by performing several tests in a test bed specifically designed for the system and collecting the output for each test.
- Parameterising the network can be done using syntactical data from several published papers that are related or similar to the system, crash reports, and transportation standards.

It was too difficult to acquire a large amount of training data for this study, as no test bed is equipped with all the sensors required for the system. No previous study provides the data required to parameterise the system. Therefore, obtaining the CPTs values in the proposed system is gathered from real high-fidelity data collected by UK DfT as crash reports dataset [1, 2, 107] and wide range of published papers and researches [48, 57, 66, 68, 112, 119-127]. As seen in figure 5.1, the network consists of 18 evidence nodes (root nodes and leaves nodes), each of which has more than two possible states. The total number of all the possible combinations of evidences is 11,289,600 possible inputs, it is impossible to illustrate all these CPTs in this chapter. Therefore, appendix A provides the full CPT's of the proposed DBN.

The following tables show three samples of the CPT of the proposed system DBN. Table 5.4 show the probabilities of the crashes depends on the day of the week [1, 2], and it illustrates the highest crash probabilities at the Wednesday and Thursday with 0.294 and 0.168 respectively.

DAY_OF_WEEK	
SUN	0.13231
MON	0.09840
TUE	0.14949
WED	0.29400
THU	0.16845
FRI	0.09891
SAT	0.05844

Table 5.4 Prior probability for DAY_OF_WEEK node

Table 5.5 shows the CPT of the driver node in the DBN. The driver node effected by the gender and age nodes as shown in Figure 5.1, to infer driver state is good or bad. For example in the second column the probability of the driver to be in bad state is higher that good driver because the age is under 18 years old.

DRIVER										
GENDER	Male					Female				
AGE	UNDER_18	BTW_19_35	BTW_36_65	BTW_66_75	OVER_75	UNDER_18	BTW_19_35	BTW_36_65	BTW_66_75	OVER_75
GOOD	0.30600	0.10350	0.37330	0.54211	0.63249	0.41494	0.69014	0.48954	0.84212	0.24164
BAD	0.69400	0.89650	0.62670	0.45789	0.36751	0.58506	0.30986	0.51046	0.15788	0.75836

Table 5.5 Prior probability for DRIVER node

Table 5.6 shows the part of the CPT of the crash node in the DBN. The crash node effected by the road type, driver, environment and the brake pedal as shown in Figure 5.1, to infer crash the severity depending on the sensors reading of the above mentioned nodes. for example in the second column the probability of the crash to be in no crash state is the higher because the brake pedal is “on”, the environment is “good” and the driver is “good” on the dual carriageway.

CRASH								
ROAD_TYPE	Dual_carriageway							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.14287	0.18513	0.12510	0.08016	0.19988	0.06905	0.08351	0.16670
Serious	0.00055	0.22210	0.12500	0.12001	0.05018	0.06903	0.12502	0.00020
Slight	0.21420	0.25918	0.37485	0.16018	0.29977	0.34471	0.33327	0.37487
NO_CRASH	0.64239	0.33359	0.37505	0.63966	0.45018	0.51721	0.45820	0.45822

Table 5. 6 Prior probability for Crash node

Appendix A provides the full CPT’s of the proposed DBN model with full details to description all the CPT’s nodes in the proposed system.

5.3.4 Inferring the hypothesis node (crash node)

In order to find the probability of the hypothesis' node according to given evidence in dynamic systems (over time), it is necessary to perform inference on a DBN. Inference is the process of combining the low-level data collected by different sensors and deducing high-level contextual information. There are several inference methods that are different from static BBNs because of the DBN property of working over time. The main goal of inference in a DBN is to calculate$(X_t|Y_{1:\tau})$. The most common types of these methods of inference are: filtering over (t), prediction over $(t+h)$, smoothing over $(t-l)$ and Viterbi decoding as showed in Section 3.5.2.3. In the present system, a prediction over $(t+h)$ is adapted to calculate $p(X_{t+h}|Y_{1:\mathrm{t}})$ and to achieve the future belief state that depends on all the given evidence from the past, where $h > 0$ and $\tau < t$. This method of inference is used to evaluate the effect of possible actions on future time. The most suitable algorithm capable of inferring the pre-crash network is the clustering algorithm, which is the most useful for small to medium-sized networks (reaching approximately three dozen nodes), and is the fastest known exact algorithm for belief-updating in BBNs.

5.4 System Learning

The proposed system has the ability to learn the CPTs values to be suitable to any country or region. The CPTs can be updated using new dataset, which must contain the same number of nodes and has at least 10,000 crash records to learn the system then get high performance.

In addition, the system has the ability to save data regarding the behaviour of the driver in the dynamic knowledge base, which can save many records as historical information for describing the driver reaction in the future hazard situations. This historical information used for updating the driver node in the proposed DBN CPTs' with the new values (learning), which reflect the driver behaviour. Learning property of the DBN can enhance the crash prediction to be more reasonable and accurate as shown in section 3.5.2.4.

5.5 Summary

This chapter presented a novel DBN to predict crashes before they happen and detect their severity level in the case of inevitable crashes. It presented the contributory factors that are chosen in the proposed system and define each node according to its group. The DBN graph with the relationship between nodes is presented too.

This chapter is also presents, how to parameterising the CPT of each node and inferring the network using a clustering algorithm to calculate the crash likelihood for the hypothesis node which was found to be the most appropriate algorithm for the proposed system. Using a DBN and taking into account a large amount of contextual information will lead to more accurate and efficient crash prediction. The final section of this chapter demonstrated the necessary steps for creating DBN using GeNIe version 2.0 software.

Chapter 6 - Pre-crash System Validation and Evaluation

Chapter Objectives:

1. Present the validity and the performance of the DBN crash prediction system.
2. Demonstrate the importance of gathering supplementary and different contexts during the inference process.
3. Present the pre-crash system Evaluation with different experiments and scenarios

6.1 Introduction

The prevention of road accidents will result in enhancing road safety and saving lives. This can done by reacting in a suitable manner to avoid or mitigate a crash. This chapter introduces the validation of the proposed DBN pre-crash system, which uses a mixture of real (The 2011 DfT dataset [1]) and synthetic data. In addition, it explains the validity and effectiveness of the proposed system in predicting crashes and their severity, by taking into account different evidence (i.e. sensor readings).

In order to validate the proposed system, it is necessary to apply all possible evidence to establish the capability of the system in predicting different kinds of crashes. As shown in Chapter 5, the system constitutes a number of nodes with different states, which lead to 11,289,600 combinations of evidence. It is not possible to examine them all in this validation and therefore combinations that reflect real live situations have been use, as will be demonstrated in the following sections. The 2011 DfT dataset [1] was used during the validation and evaluation. This contains 150,000 crash records. A number of different scenarios have been present in order to demonstrate the comprehensive nature of the system and to illustrate its ability to predict different levels of crash severity. Examining a large number of combinations of evidence and presenting different scenarios will challenge the system, demonstrate that it is sufficiently comprehensive, and establish its validity when predicting different crash situations.

During the validation and evaluation three time slices have been used to predict the crash state: time slice *(t-1)* (previous), time slice *(t) (*current) and future time *(t+1)*. The inclusion of three time slices ensures that the temporal aspect of the system has been taken in to account in addition to the sensor readings.

Firstly, the results of inputting several combinations of evidence will be reveal. Secondly, different experiments will be present. The validation results will demonstrate the system's ability to predict different levels of severity accurately. Moreover, the experiments presented in this chapter will evaluate and validate the capabilities of the system in predicting a crash while driving.

6.2 System validation using synthetic data

This validation demonstrates the effect of the root (information) nodes on their children (i.e. the effect of the day of week; light, road surface and weather conditions; time; traffic status; gender and age on the environment, driver and road type nodes). Before their effect has be shown, it is important to explain the root node, as shown in Tables 6.1 – 6.2. As shown in chapter Five, the DBN graph in Figure 5.1 demonstrates the relationship between the DBN nodes, along with a detailed explanation to clarify their relationships (i.e. root, parents, and children). The tables below demonstrate the effect of the root nodes on the crash parents' nodes (environment, driver and road type). Since the road type node has no parent, it will be consider an evidence node.

Table 6.1 illustrates the root nodes that affect the environment nodes.

No.	Information Nodes Factors
1	DAY_OF_WEEK
2	LIGHT_CONDITIONS
3	ROAD_SURFACE_CONDITIONS
4	TIME
5	TRAFFIC_STATUS
6	WEATHER_CONDITIONS

Table 6.1 The factors that affect the Environment node

Table 6.2 illustrates the root nodes that affect the driver node.

No.	Information Nodes Factors
1	GENDER
2	AGE

Table 6.2 The factors that affect the Driver node

Table 6.3 constitutes the states of the root nodes (traffic status, light conditions, weather conditions and road surface condition) that affect the environment node, the table shows the inference results of belief and state of the environment node according to the inputted evidence. The total number of combinations is 140 evidence.

No.	TRAFFIC STATUS	LIGHT CONDITIONS	WEATHER CONDITIONS	ROAD SURFACE CONDITIONS	ENVIRONMENT Belief and state
1	DENSITY	Daylight	Fine_AND_high_winds	Dry	0.58, GOOD
2	DENSITY	Daylight	Fine_AND_high_winds	Flood_over_3cm_deep	0.62, GOOD
3	DENSITY	Daylight	Fine_AND_high_winds	Frost_or_ice	0.83, BAD
4	DENSITY	Daylight	Fine_AND_high_winds	Snow	0.56, BAD
5	DENSITY	Daylight	Fine_AND_high_winds	Wet_or_damp	0.58, GOOD
6	DENSITY	Daylight	Fine_NO_high_winds	Dry	0.52, GOOD
7	DENSITY	Daylight	Fine_NO_high_winds	Flood_over_3cm_deep	0.55, BAD
8	DENSITY	Daylight	Fine_NO_high_winds	Frost_or_ice	0.51, GOOD
9	DENSITY	Daylight	Fine_NO_high_winds	Snow	0.54, BAD
10	DENSITY	Daylight	Fine_NO_high_winds	Wet_or_damp	0.58, BAD
11	DENSITY	Daylight	Fog_or_mist	Dry	0.53, BAD
12	DENSITY	Daylight	Fog_or_mist	Flood_over_3cm_deep	0.74, BAD
13	DENSITY	Daylight	Fog_or_mist	Frost_or_ice	0.66, BAD
14	DENSITY	Daylight	Fog_or_mist	Snow	0.58, GOOD
15	DENSITY	Daylight	Fog_or_mist	Wet_or_damp	0.86, BAD
16	DENSITY	Daylight	Raining_AND_high_winds	Dry	0.56,GOOD
17	DENSITY	Daylight	Raining_AND_high_winds	Flood_over_3cm_deep	0.85, GOOD
18	DENSITY	Daylight	Raining_AND_high_winds	Frost_or_ice	0.59, BAD
19	DENSITY	Daylight	Raining_AND_high_winds	Snow	0.61,BAD
20	DENSITY	Daylight	Raining_AND_high_winds	Wet_or_damp	0.7, BAD
21	DENSITY	Daylight	Raining_NO_high_winds	Dry	0.78,BAD
22	DENSITY	Daylight	Raining_NO_high_winds	Flood_over_3cm_deep	0.77, BAD
23	DENSITY	Daylight	Raining_NO_high_winds	Frost_or_ice	0.8, GOOD
24	DENSITY	Daylight	Raining_NO_high_winds	Snow	0.79, GOOD
25	DENSITY	Daylight	Raining_NO_high_winds	Wet_or_damp	0.74, BAD
⋮	⋮	⋮	⋮	⋮	⋮
140	NORMAL	Darkness	Snowing_NO_high_winds	Wet_or_damp	0.68, BAD

Table 6.3 The information nodes that are affected by the environment node and its belief.

Table 6.4 constitutes the states of the root nodes (gender and age) that affect the driver node, and shows the inference results of the belief and state of the driver node according to the inputted evidence. The total number of combinations is 10 evidence.

	GENDER	AGE	DRIVER Belief and state
1	Male	UNDER_18	0.69, BAD
2	Male	BTW_19_35	0.90, BAD
3	Male	BTW_36_65	0.63, BAD
4	Male	BTW_66_75	0.45, GOOD
5	Male	OVER_75	0.63, GOOD
6	Female	UNDER_18	0.59, BAD
7	Female	BTW_19_35	0.69¸GOOD
8	Female	BTW_36_65	0.51¸BAD
9	Female	BTW_66_75	0.84¸GOOD
10	Female	OVER_75	0.76, BAD

Table 6.4 The information nodes that are affected the driver node and its belief.

Since the road type has no parents, it will be consider an evidence node. Therefore, the nodes that will affect the crash nodes are environment, driver and road type.

Having considered the parents of crash node as evidence nodes, this step of the validation presents the effect of environment, driver and road type, in addition to the effect of observation nodes on the crash node. The environment, driver and road type nodes will be treat as evidence nodes throughout this step of evaluation. After considering the environment, driver, road type, brake pedal, speed, distance, and lane nodes as evidence nodes, the total number of all possible combinations of evidence attained to 1152 possible cases.

During the evaluation, three time slices have been used (previous *(t-1)*, current *(t)* and future time slices *(t+1)*) in order to include the evidence from the previous and current time slices to predict the state in the future time slice. During the inference, the hypothesis node (crash) at time slice (*t+1*) will be influenced by all the information and observable nodes at time slice (*t*), as well as by the hypothesis node at time slice (*t-1*).

Table 6.5 shows all the possible combinations of evidence for the parents of the crash node, which results in 24 possible combinations.

Combination No.	Road Type	Driver	Environment
1	Dual_carriageway	GOOD	GOOD
2	Dual_carriageway	GOOD	BAD
3	Dual_carriageway	BAD	GOOD
4	Dual_carriageway	BAD	BAD
5	One_way_street	GOOD	GOOD
6	One_way_street	GOOD	BAD
7	One_way_street	BAD	GOOD
8	One_way_street	BAD	BAD
9	One_way_street_Slip_road	GOOD	GOOD
10	One_way_street_Slip_road	GOOD	BAD
11	One_way_street_Slip_road	BAD	GOOD
12	One_way_street_Slip_road	BAD	BAD
13	Roundabout	GOOD	GOOD
14	Roundabout	GOOD	BAD
15	Roundabout	BAD	GOOD
16	Roundabout	BAD	BAD
17	Single_carriageway	GOOD	GOOD
18	Single_carriageway	GOOD	BAD
19	Single_carriageway	BAD	GOOD
20	Single_carriageway	BAD	BAD
21	Slip_road	GOOD	GOOD
22	Slip_road	GOOD	BAD
23	Slip_road	BAD	GOOD
24	Slip_road	BAD	BAD

Table 6.5 The potential cases of road type, driver and environment .

With each combination for the above table, there is a further combination of evidence from the observation nodes, i.e. speed, distance, lane, and brake pedal nodes. Each input set will therefore include a combination of environment, driver, road type, and observation nodes, as shown in Tables 6.6 to 6.9.

The inference (prediction) process in DBN is used to find the belief in the hypothesis node at the future time slice (*t+1*), taking into account information and observation nodes from time slice (*t*), as well as time slice (*t-1*) [112]. Therefore, in this step of evaluation, the nodes evident at (*t-1*) and (*t*) has been inputted to establish belief in the crash node at time slice (*t+1*).

In the following tables 6.6 to 6.9, inputting different combinations of evidence have demonstrated the confidence in the crash node. As shown in Table 6.5, setting the state of the crash's parent nodes to specific states, and inputting a combination of evidence from the children nodes, will challenge the system and ensure that it is sufficiently accurate in predicting different kinds of crash states with different sensor readings.

Table 6.6 shows the degree of confidence in predicting a crash and its severity by setting the environment, driver and road type the combination number 1 in Table 6.5 (road type = dual carriageway, driver = good and environment = good). In addition, all possible combinations of evidence of observation nodes inputted. However, the inputted combinations that include the same lane state (for lane node) are included, as there is no crash if there is a separate lane. In addition, the combinations of evidence where the brake pedal state was 'on' in time slice (*t-1*) and 'off' in time slice (*t*) excluded. Moreover, the cases where the brake pedal was 'off' in time slice (*t-1)* and 'on' in time slice (*t*) excluded. The reason behind removing the above-mentioned combinations is that they do not reflect real world situations. Therefore, the total number attained is 24 combinations, as shown in Table 6.6. This situation will be the same in the following tables.

Combination 1 in Table 6.5 (Road type= dual_carriageway, driver=good and environment=good)										
t-1				***T***			***t+1***			
BRAKE_PEDAL		SPEED	DISTANCE	BRAKE_PEDAL	SPEED	DISTANCE	CRASH_STATE			
							Fatal	Serious	Slight	No_crash
1	ON	OVER_LIMIT	less20m	ON	LIMIT	less20m	0.00	0.00	0.13	0.87
2	ON	OVER_LIMIT	less5m	ON	LIMIT	less5m	0.04	0.26	0.45	0.25
3	ON	OVER_LIMIT	less1m	ON	LIMIT	less1m	0.24	0.49	0.22	0.05
4	ON	LIMIT	less20m	ON	BELOW_LIMIT	less20m	0.00	0.00	0.13	0.87
5	ON	LIMIT	less5m	ON	BELOW_LIMIT	less5m	0.01	0.12	0.45	0.43
6	ON	LIMIT	less1m	ON	BELOW_LIMIT	less1m	0.04	0.22	0.44	0.30
7	ON	BELOW_LIMIT	less20m	ON	Zero	less20m	0.00	0.00	0.13	0.87
8	ON	BELOW_LIMIT	less5m	ON	Zero	less5m	0.00	0.00	0.15	0.85
9	ON	BELOW_LIMIT	less1m	ON	Zero	less1m	0.01	0.01	0.16	0.83
10	ON	Zero	less20m	ON	Zero	less20m	0.00	0.00	0.13	0.87
11	ON	Zero	less5m	ON	Zero	less5m	0.00	0.00	0.14	0.86
12	ON	Zero	less1m	ON	Zero	less1m	0.00	0.00	0.14	0.86
13	OFF	OVER_LIMIT	less20m	OFF	OVER_LIMIT	less5m	0.16	0.64	0.20	0.00
14	OFF	OVER_LIMIT	less5m	OFF	OVER_LIMIT	less1m	0.78	0.13	0.04	0.05
15	OFF	OVER_LIMIT	less1m	OFF	OVER_LIMIT	less1m	0.81	0.10	0.03	0.06
16	OFF	LIMIT	less20m	OFF	LIMIT	less5m	0.04	0.26	0.28	0.42
17	OFF	LIMIT	less5m	OFF	LIMIT	less1m	0.10	0.53	0.28	0.10
18	OFF	LIMIT	less1m	OFF	LIMIT	less1m	0.10	0.53	0.28	0.09
19	OFF	BELOW_LIMIT	less20m	OFF	BELOW_LIMIT	less5m	0.00	0.05	0.34	0.61
20	OFF	BELOW_LIMIT	less5m	OFF	BELOW_LIMIT	less1m	0.00	0.08	0.47	0.45
21	OFF	BELOW_LIMIT	less1m	OFF	BELOW_LIMIT	less1m	0.00	0.08	0.49	0.43
22	OFF	Zero	less20m	OFF	Zero	less5m	0.00	0.00	0.14	0.86
23	OFF	Zero	less5m	OFF	Zero	less1m	0.00	0.00	0.15	0.85
24	OFF	Zero	less1m	OFF	Zero	less1m	0.00	0.00	0.15	0.85

Table 6.6 The belief in crash node by setting the environment, driver and road type, as in combination 1 in Table 6.5

Table 6.6 illustrate twelve combinations of evidence have been inputted to demonstrate the system's ability to predict the likelihood of a crash. The status of the brake pedal node has been set to 'on' in time slices *(t-1)* and *(t)*.

As can be seen from the Table 6.6, combinations 1-3 (where the speed is over the limit) different states of distance node leads to different conclusions in the crash node. The conclusion in the crash node is 0.87 no crash, 0.45 slight and 0.49 serious with distance of less than 20m, less than 5m and less than 1m respectively. From combinations 4-6 the speed is within the limit. Different states of crash nodes have been reached: 0.87 no crash, 0.45 slight, and 0.44 slight, with distance of less than 20m, less than 5m and less than 1m respectively. While in combinations 7-12 (where the speed is below the limit or zero), the no crash state has found as the highest. The above table demonstrates the effect of speed and distance in predicting a crash and its level of severity where the brake pedal is 'on'.

The combinations 13 to 24 – Table 6.6 denoting the states of the brake pedal node has been set to "off" in time slices *(t-1)* and *(t)*. It can be seen from the Table 6-6, combination 13 of evidence belief in serious state was found to be 0.64 where the distance was lee20m and the speed was over limit in time slice *(t-1)*. In time slice *(t),* the distance become less5m and speed stayed on over limit because the brake pedal was "off". While in combinations 14-15 the state changed to fatal where the distance was less1m and speed was over limit. In combination 16 the belief in crash node was 0.42 no crash because the speed was limit and the distance less5m. in combinations 17-18, the speed was limit and the distance was less1m. therefore, the belief in serious was 0.53. in combinations 19-21 the speed was below limit while the brake pedal was "off", different state of distance node lead to different degree of belief in crash node 0.61 no crash, 0.47 slight and 0.49 slight with less5m, less1m and less1m respectively. Again this shows the validity of our proposed system.

Table 6.7 shows the belief in crash node by setting the environment, driver and road type as in combination number 2 in Table 6.5 (road type= Dual_carriageway, driver=good and Environment=bad) . And all possible combinations of evidence of observation nodes has been inputted.

Combination 2 in Table 6.5 (road type= Dual_carriageway, driver=good and Environment=bad)										
t-1				***T***			***t+1***			
BRAKE_PEDAL		SPEED	DISTANCE	BRAKE_PEDAL	SPEED	DISTANCE	CRASH_STATE			
							Fatal	Serious	Slight	No_crash
1	ON	OVER_LIMIT	less20m	ON	LIMIT	less20m	0.00	0.05	0.19	0.77
2	ON	OVER_LIMIT	less5m	ON	LIMIT	less5m	0.29	0.42	0.12	0.18
3	ON	OVER_LIMIT	less1m	ON	LIMIT	less1m	0.52	0.35	0.08	0.05
4	ON	LIMIT	less20m	ON	BELOW_LIMIT	less20m	0.00	0.05	0.19	0.77
5	ON	LIMIT	less5m	ON	BELOW_LIMIT	less5m	0.05	0.49	0.14	0.32
6	ON	LIMIT	less1m	ON	BELOW_LIMIT	less1m	0.19	0.45	0.13	0.23
7	ON	BELOW_LIMIT	less20m	ON	Zero	less20m	0.00	0.04	0.19	0.77
8	ON	BELOW_LIMIT	less5m	ON	Zero	less5m	0.00	0.08	0.18	0.74
9	ON	BELOW_LIMIT	less1m	ON	Zero	less1m	0.01	0.09	0.18	0.72
10	ON	Zero	less20m	ON	Zero	less20m	0.00	0.04	0.19	0.77
11	ON	Zero	less5m	ON	Zero	less5m	0.00	0.05	0.19	0.76
12	ON	Zero	less1m	ON	Zero	less1m	0.00	0.05	0.19	0.76
13	OFF	OVER_LIMIT	less20m	OFF	OVER_LIMIT	less5m	0.79	0.20	0.01	0.00
14	OFF	OVER_LIMIT	less5m	OFF	OVER_LIMIT	less1m	0.78	0.20	0.01	0.00
15	OFF	OVER_LIMIT	less1m	OFF	OVER_LIMIT	less1m	0.80	0.20	0.01	0.00
16	OFF	LIMIT	less20m	OFF	LIMIT	less5m	0.16	0.32	0.14	0.37
17	OFF	LIMIT	less5m	OFF	LIMIT	less1m	0.38	0.41	0.10	0.11
18	OFF	LIMIT	less1m	OFF	LIMIT	less1m	0.38	0.41	0.10	0.10
19	OFF	BELOW_LIMIT	less20m	OFF	BELOW_LIMIT	less5m	0.00	0.33	0.16	0.51
20	OFF	BELOW_LIMIT	less5m	OFF	BELOW_LIMIT	less1m	0.00	0.47	0.15	0.37
21	OFF	BELOW_LIMIT	less1m	OFF	BELOW_LIMIT	less1m	0.00	0.49	0.15	0.36
22	OFF	Zero	less20m	OFF	Zero	less5m	0.00	0.06	0.19	0.76
23	OFF	Zero	less5m	OFF	Zero	less1m	0.00	0.07	0.18	0.75
24	OFF	Zero	less1m	OFF	Zero	less1m	0.00	0.07	0.18	0.75

Table 6.7 The belief in crash node by setting the environment, driver and road type, as in combinations 2 in Table 6.5

Table 6.7 including combinations 1 to 12, the status of the brake pedal node has been set to ‘on’ in time slices *(t-1)* and *(t)*. It is can be seen from the combinations 1-3, where the speed is over the limit, different states of distance node lead to different beliefs in the crash node. The belief in the crash node of less than 20m, less than 5m and less than 1m respectively. From combinations 4-6 the speed has been within the limit, different states of crash nodes have been reached: 0.77 no crash, 0.49 serious, and 0.45 serious with distance of less than 20m, less than 5m and less than 1m respectively. While in combinations 7-12, where the speed is below the limit or zero, the belief in the no crash state has found as the highest. The above chart demonstrates the effect of the speed and distance in predicting the crash and its level of severity where the brake pedal is ‘on’.

The combinations 13 to 24 – Table 6.7 denoting the states of crash node. the brake pedal node has been set to “off” in time slices (*t-1)* and (*t)*. It can be seen from the Table 6.7, with combination 13-15 of evidence the belief in fatal state was found to be the highest one in different distance and the speed was over limit in time slice *(t-1)*. In time slice *(t)*, the distance decreased and speed stayed on over limit because the brake pedal was “off”. In combination 16 the belief in crash node was 0.37 no crash because the speed was limit and the distance less20m. in combinations 17-18, the belief in crash node was 0.41 serious because the speed was limit and the distance less1m at time slice *(t)*. In combinations 19 the belief in crash node was 0.51 no crash because the speed was below limit and the distance less20m. In combinations 20-21 the speed was below limit while the brake pedal was “off”, different state of distance node lead to different degree of belief in crash node was 0.47 and 0.49 serious because the speed was below limit and the distance less than 1m at time slice *t*. while, in combinations 22-24 the belief in crash node was no crash because the speed was zero. Again this shows the validity of our proposed system.

Table 6.8 shows the belief in crash node by setting the environment, driver and road type as in combination number 3 in Table 6.5 (road type= Dual_carriageway, driver=bad and Environment=good). And all possible combinations of evidence of observation nodes has been inputted.

#	*t-1*			*t*			*t+1*			
	BRAKE_ PEDAL	SPEED	DISTANCE	BRAKE_PEDAL	SPEED	DISTANCE	CRASH_STATE			
							Fatal	Serious	Slight	No_crash
1	ON	OVER_LIMIT	less20m	ON	LIMIT	less20m	0.00	0.05	0.24	0.72
2	ON	OVER_LIMIT	less5m	ON	LIMIT	less5m	0.25	0.45	0.14	0.16
3	ON	OVER_LIMIT	less1m	ON	LIMIT	less1m	0.52	0.31	0.12	0.05
4	ON	LIMIT	less20m	ON	BELOW_LIMIT	less20m	0.00	0.05	0.24	0.72
5	ON	LIMIT	less5m	ON	BELOW_LIMIT	less5m	0.05	0.52	0.14	0.28
6	ON	LIMIT	less1m	ON	BELOW_LIMIT	less1m	0.18	0.49	0.14	0.20
7	ON	BELOW_LIMIT	less20m	ON	Zero	less20m	0.00	0.05	0.24	0.72
8	ON	BELOW_LIMIT	less5m	ON	Zero	less5m	0.00	0.08	0.23	0.69
9	ON	BELOW_LIMIT	less1m	ON	Zero	less1m	0.01	0.10	0.22	0.67
10	ON	Zero	less20m	ON	Zero	less20m	0.00	0.05	0.24	0.72
11	ON	Zero	less5m	ON	Zero	less5m	0.00	0.05	0.23	0.71
12	ON	Zero	less1m	ON	Zero	less1m	0.00	0.06	0.23	0.71
13	OFF	OVER_LIMIT	less20m	OFF	OVER_LIMIT	less5m	0.83	0.06	0.04	0.07
14	OFF	OVER_LIMIT	less5m	OFF	OVER_LIMIT	less1m	0.83	0.06	0.04	0.07
15	OFF	OVER_LIMIT	less1m	OFF	OVER_LIMIT	less1m	0.84	0.05	0.04	0.08
16	OFF	LIMIT	less20m	OFF	LIMIT	less5m	0.16	0.33	0.17	0.33
17	OFF	LIMIT	less5m	OFF	LIMIT	less1m	0.39	0.40	0.14	0.07
18	OFF	LIMIT	less1m	OFF	LIMIT	less1m	0.39	0.40	0.14	0.07
19	OFF	BELOW_LIMIT	less20m	OFF	BELOW_LIMIT	less5m	0.00	0.36	0.18	0.46
20	OFF	BELOW_LIMIT	less5m	OFF	BELOW_LIMIT	less1m	0.00	0.49	0.15	0.35
21	OFF	BELOW_LIMIT	less1m	OFF	BELOW_LIMIT	less1m	0.00	0.51	0.15	0.34
22	OFF	Zero	less20m	OFF	Zero	less5m	0.00	0.06	0.23	0.71
23	OFF	Zero	less5m	OFF	Zero	less1m	0.00	0.07	0.23	0.70
24	OFF	Zero	less1m	OFF	Zero	less1m	0.00	0.07	0.23	0.70

Table 6.8 The belief in crash node by setting the environment, driver and road type, as in combination 3 in Table 6.5

The combination 1 to 12 – Table 6.8) the first twelve combinations of evidence are have been inputted to demonstrate the system's ability to predict the likelihood of a crash. The status of the brake pedal node has been set to 'on' in time slices *(t-1)* and *(t)*. It can be seen from the chart that combination 1-3, where the speed is over limit, different states of distance node lead to different belief in the crash node. The crash node is 0.87 no crash, 0.45 slight and 0.49 serious with distance of less than 20m, less than 5m and less than 1m respectively. From combination 4-6, the speed is within the limit, and different states of crash nodes have reached 0.87 no crash, 0.45 slight, and 0.44 slight with distance of less than 20m, less than 5m and less than 1m respectively. In combination 7-12, where the speed is below the limit or zero, the conclusion of a no crash state has found as the highest. The above chart demonstrates the effect of speed and distance in predicting the crash and its level of severity where the brake pedal is ‘on’.

The combination 13 to 24 – Table 6.8 the brake pedal node has been set to “off” in time slices *(t-1)* and *(t)*. It can be seen from the Table 6.8, with combination 13-15 the belief in fatal state was found to be the highest one in different distance and the speed was over limit in time slice *(t-1)*. In time slice *(t)*, the distance decreased and speed stayed on over limit because the brake pedal was “off”. In combination 16 the belief in crash node was 0.33 no crash because the speed was limit and the distance less20m. in combination 17-18, the belief in crash node was 0.40 serious because the speed was limit and the distance less1m at time slice t. In combination 19 the belief in crash node was 0.49 no crash because the speed was below limit and the distance less20m. In combination 20-21 the speed was below limit while the brake pedal was “off”, different state of distance node lead to different degree of belief in crash node was 0.49 and 0.51 serious because the speed was below limit and the distance less than 1m at time slice t. while, in combination 22-24 the belief in crash node was no crash because the speed was zero.

Table 6.9 shows a comparison between the degree of belief in the crash nodes with three different situations of parent node (Driver, Environment and road type). The comparison include the highest degrees of belief from Table 6.6 to 6.8 which are concluded in Table 6.9.

#	Deriver=GOOD Environment=GOOD Road type= Dual carriageway				Deriver=BAD Environment=GOOD Road type= Dual carriageway				Deriver=GOOD Environment=BAD Road type= Dual carriageway			
	Crash				Crash				Crash			
	Fatal	Serous	Slight	No_crash	Fatal	Serous	Slight	No_crash	Fatal	Serous	Slight	No_crash
1	0.00	0.00	0.13	**0.87**	0.00	0.05	0.19	**0.77**	0.00	0.05	0.24	**0.72**
2	0.04	0.26	**0.45**	0.25	0.29	**0.42**	0.12	0.18	0.25	**0.45**	0.14	0.16
3	0.24	**0.49**	0.22	0.05	**0.52**	0.35	0.08	0.05	**0.52**	0.31	0.12	0.05
4	0.00	0.00	0.13	**0.87**	0.00	0.05	0.19	**0.77**	0.00	0.05	0.24	**0.72**
5	0.01	0.12	**0.45**	0.43	0.05	**0.49**	0.14	0.32	0.05	**0.52**	0.14	0.28
6	0.04	0.22	**0.44**	0.30	0.19	**0.45**	0.13	0.23	0.18	**0.49**	0.14	0.20
7	0.00	0.00	0.13	**0.87**	0.00	0.04	0.19	**0.77**	0.00	0.05	0.24	**0.72**
8	0.00	0.00	0.15	**0.85**	0.00	0.08	0.18	**0.74**	0.00	0.08	0.23	**0.69**
9	0.01	0.01	0.16	**0.83**	0.01	0.09	0.18	**0.72**	0.01	0.10	0.22	**0.67**
10	0.00	0.00	0.13	**0.87**	0.00	0.04	0.19	**0.77**	0.00	0.05	0.24	**0.72**
11	0.00	0.00	0.14	**0.86**	0.00	0.05	0.19	**0.76**	0.00	0.05	0.23	**0.71**
12	0.00	0.00	0.14	**0.86**	0.00	0.05	0.19	**0.76**	0.00	0.06	0.23	**0.71**
13	0.16	**0.64**	0.20	0.00	**0.79**	0.20	0.01	0.00	**0.83**	0.06	0.04	0.07
14	**0.78**	0.13	0.04	0.05	**0.78**	0.20	0.01	0.00	**0.83**	0.06	0.04	0.07
15	**0.81**	0.10	0.03	0.06	**0.80**	0.20	0.01	0.00	**0.84**	0.05	0.04	0.08
16	0.04	0.26	0.28	**0.42**	0.16	0.32	0.14	**0.37**	0.16	**0.33**	0.17	0.33
17	0.10	**0.53**	0.28	0.10	0.38	**0.41**	0.10	0.11	0.39	**0.40**	0.14	0.07
18	0.10	**0.53**	0.28	0.09	0.38	**0.41**	0.10	0.10	0.39	**0.40**	0.14	0.07
19	0.00	0.05	0.34	**0.61**	0.00	0.33	0.16	**0.51**	0.00	0.36	0.18	**0.46**
20	0.00	0.08	**0.47**	0.45	0.00	**0.47**	0.15	0.37	0.00	**0.49**	0.15	0.35
21	0.00	0.08	**0.49**	0.43	0.00	**0.49**	0.15	0.36	0.00	**0.51**	0.15	0.34
22	0.00	0.00	0.14	**0.86**	0.00	0.06	0.19	**0.76**	0.00	0.06	0.23	**0.71**
23	0.00	0.00	0.15	**0.85**	0.00	0.07	0.18	**0.75**	0.00	0.07	0.23	**0.70**
24	0.00	0.00	0.15	**0.85**	0.00	0.07	0.18	**0.75**	0.00	0.07	0.23	**0.70**

Table 6.9 a comparism between the degree of belief in the crash nodes with three different situations

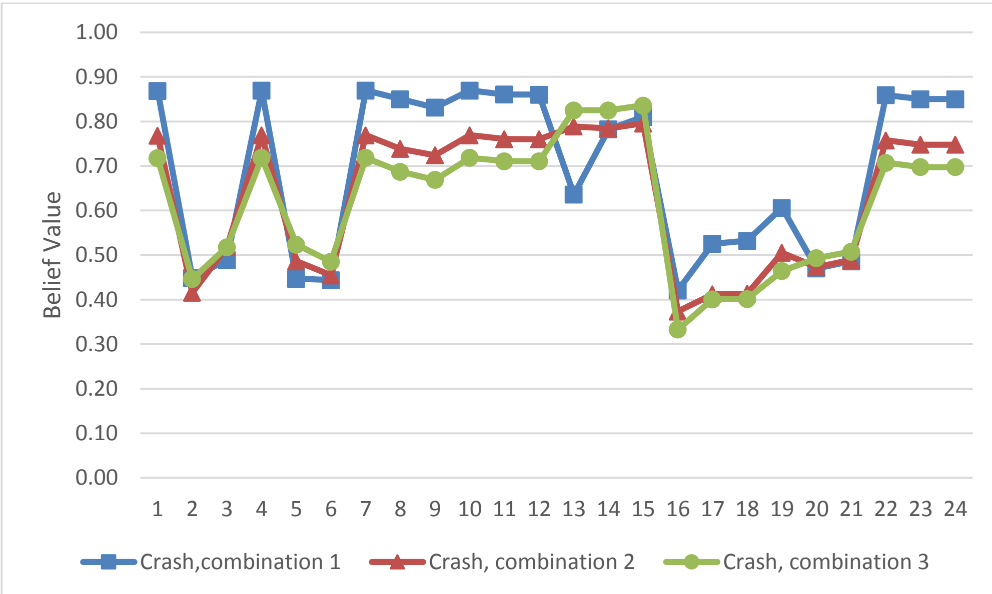

Figure 6.1 a comparison between the degree of belief in the crash nodes with three different situations

Figure 6.1 shows a comparison between the degrees found in the crash nodes with three different situations of parent node (driver, environment and road type). The comparison includes the highest degrees of belief from Table 6.6 to 6.8. As shown in Figure 6.1, each point on the x-axis represents three combinations of evidence, where the observable node states are the same, while the changes taken place in the state of the parent nodes. At the first point, the degree of belief in no crash reaches to the highest, due to the effect of a good driver and good environment. In point two, the system predicts a slight crash where the environment is good, while the state changed to serious when the environment is bad and the driver is good, or when the environment is good and the driver is bad. In the third point, the system predicts a serious crash where the environment is good, while the state changed to fatal when the environment is bad and the driver is good, or when the environment is good and the driver is bad. In the fourth point, the no crash predicted in all three situations, but the highest value was when the environment and the driver are good. In the fifth and sixth points, the system predicts a slight crash then changes to serious, due to the effect of a bad environment or driver.

From points 7 to 12, the system predicts no crash, but the highest value is when the environment and the driver are good. In point 13, the prediction is a serious crash when the environment and driver are good, but it changes to be a fatal crash when any of the environment and/or driver combinations change to a bad state. In points 14 and 15 the system predicts a fatal crash in all three situations, but with different degrees of certainty, where the highest degree when one or both of the driver/environment states changes to bad. In point 15, no crash is predicted when the driver is perceived as good, but when the driver changes to bad the system prediction is changed to be a serious crash. In point, 17 and 18 the prediction is a serious crash in all three situations but with little difference in the degree of belief. In point 19, no crash predicted in all three situations, but the highest value is when the environment and the driver are good. In points, 20 and 21 the system predicts a slight crash, and then changes to serious due to the effect of the bad environment or driver. From point 22 to 24 the system predicts no crash where expected, but the highest value is when the environment and the driver are good.

As demonstrated by the above table, the system is able to predict the state of a crash (fatal, serious, slight and no crash) accurately over time by applying different sensor readings (including parents and children). This proves the validity and accuracy of the proposed system for predicting crash occurrences and their severity, where different combinations of inputs lead to different states with different degrees of belief.

6.3 System Evaluation

This evaluation present different experiment, during the following experiments, the validity and efficiency of the proposed DBN crash prediction model in predicting different levels of crash severity will demonstrated. It is difficult to predict the crash from one perspective (i.e. focusing solely on the driver or vehicle), as this may not describe the complete situation on the road. Therefore, information related to the driver, the vehicle and the environment needs to be taken in to account to predict the likelihood of a crash accurately. In this section, the ability of the proposed system to predict different levels of severity, (fatal, serious, slight and no crash) will established.

DBN can be used to reason or present information concerning a specific domain. In the reasoning process, a group of nodes are instantiated according to the input data. This is followed by a calculation of their effect on the probability of other variables (hypothesis nodes) [4]. In this evaluation, the inference (i.e. prediction) process takes place at every time slice where the sensors collect the inputs in the previous time slice (*t-1*) and the current time slice (*t*) in order to predict the future time slice (*t+1*). Each time slice represents a part of the developing system (crash). This can explained as a known period of time, in which the system receives the sensors readings and feeds them into the system.

Since there was no testbed and a syntactical data were use in this evaluation, it is difficult to decide the exact time of each time slice. This has therefore assumed to be one second, which can be more or less depending on the real time needed for the collection, transfer and feeding of the data into the system. Road crashes (i.e. fatal, serious and slight) cause a high number of fatalities, and therefore predicting the crash sufficiently early will lead to a reduction in the number of fatalities and thus save lives. The system concentrates on predicting the above-mentioned types of crashes.

The greatest challenge in crash prediction is in its level of severity, due to the complexity of factors that lead to a decision be made. A reliable system should therefore accurately decide the severity level of a crash. The following sections provide an explanation concerning the proposed system's ability to predict the vehicle crash severity with different scenarios. The resulting prediction must aid the driver or vehicle to avoid the crash by any suitable driver action, or by applying an avoiding application to prevent the crash. The input scenarios in Tables 6.10 – 6.12 are showed the effect of the nodes that have a direct arrow to the hypothesis node (Crash), this is to reduce the experiment complexity and understandable.

6.3.1 Experiment 1: Predicting a fatal crash

A crash is consider fatal crash if the following criteria are meet, but is not limited to them.

- The speed is over limit.
- The distance is less than one metre.
- The lane is the same lane.
- The brake pedal is off.
- The likelihood of a fatal crash may increase if the environment and the driver are bad.

The following scenario will demonstrate the validity of the proposed system in predicting a fatal crash. As shown in Figure 6.8, the vehicle is moving from point A to point B on a dual carriageway. The period, in which the vehicle is moving between these points, are divided into 6 equivalent time slices, with the 6^{th} representing future time *(t+1)*. Each time slice represents a period of one second during which the vehicle collects new information via sensors and feeds it into the system.

The inference (prediction) process is carry out every second *(t)*, depending on the sensed information from *(t-1)* and *(t)* to predict the situation in time slice *(t+1)*. It is can be seen from the figure that the vehicles with the one ahead are both travelling in the same lane. From time slices, 1-3 the following criteria presented, the brake pedal is off, the distance is more than 20 metres, the speed is over limit and the lane is same. In all these inputs, there is no indication of any crash. Therefore, the system will not take place in all time slices (1-3). From time slice 4, the distance has changed to be less than 20 metres, while the other criteria remain the same. In time slice 5, the distance has changed to less than 5 metres due to the fact that the brake pedal is still off and the speed is over the limit. The inference process will therefore take place in time slice 5, which is the current time *(t),* in order to predict the crash at future time *(t+1)* (i.e. time slice 6).

Modelling the above scenario using the proposed system could undertake by setting the states of the nodes according to the data provided by the sensors. Table 6.10 illustrates the combination of evidence in time slices 1-5. The system will predict the crash state at time slice 6.

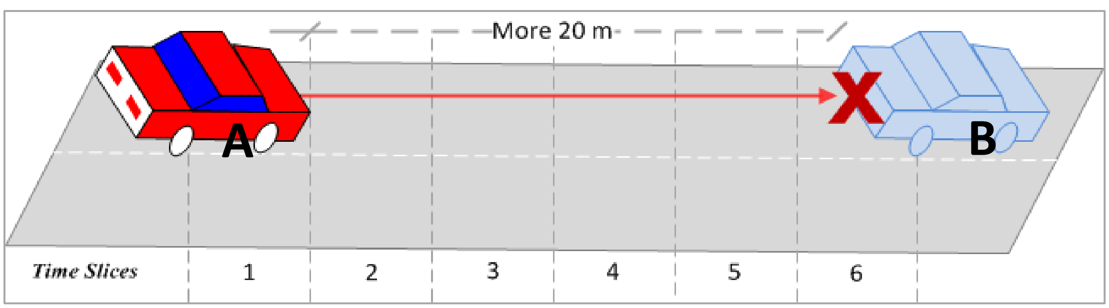

Figure 6.2 Vehicle is moving from point A to point B on a dual carriageway road

	Time Slices	***1***	***2***	***3***	***4 (t-1)***	***5 (t)***	***6 (t+1)***
Scenario Inputs	Brake_ Pedal	OFF	OFF	OFF	OFF	OFF	To be predicted
	Distance	more20m	more20m	more20m	Less20m	Less5m	To be predicted
	Speed	Over_limit	Over_limit	Over_limit	Over_limit	Over_limit	To be predicted
	Lane	Same	Same	Same	Same	Same	To be predicted
Crash belief and state of (t+1) time slice							
Fatal		-	-	-	-	-	**1.00**
Serious		-	-	-	-	**0.83**	-
Slight		-	-	-	-	-	-
No crash		**0.97**	**0.97**	**0.97**	**0.84**	-	-

Table 6.10 The experiment and the crash inference results

Figure 6.2 demonstrates the inference results of the proposed DBN model after receiving the sensors reading in time slices 1-5, in order to predict the crash at time slice 6. It is can be seen from Table 6.10 that the system has predicted no crash at time slice 1-3, with a belief of approximately 0.97. This is because all the evidence from time slice 1 to 3 indicates that the distance is more than 20 metres and that this is an indicator of no crash to the system, due to the fact that the distance is over the critical distance. At time slice 4, the distance is change to be less than 20 metres, which is an indication to the DBN model to begin the inferring process to find the crash node degree of belief to predict the crash. The prediction is 0.84 no crash, due to the effect of being over the speed limit and the brake pedal being off. At time slice 5 (i.e. current time), the prediction is 0.83 serious, as the distance is changed to less than 5 metres, the speed and brake pedal remaining the same. Finally, the system predicts the belief degree is 1.00 fatal crash at time slice 6 (future time) as the system predicts that the distance will be less than 1 metre. The speed and the brake pedal will remain the same. This result of prediction must aid the driver or the vehicle to avoid the crash by undertaking an action or applying an avoiding application in order to avoid the crash or mitigate its severity.

6.3.2 Experiment 2: Predicting the serious crash

The crash considered serious crash if the following criteria are meet, but not limited to them.

- The speed is within the road limit range.
- The distance is less than one metre.
- The lane is the same lane.
- The brake pedal is off.
- The likelihood of a serious crash might increase if the environment and the driver are bad.

The following scenario will demonstrate the validity of the proposed system in predicting a serious crash. As demonstrated in Figure 6.3, the vehicle is moving from point A to point B on a dual carriageway. Figure 6.3 shows that the vehicle and the one ahead are travelling in the same lane. From time slice, 1-3 the following criteria are present: the brake pedal is ‘off’, the distance is more 20 metres, the speed is within the limit and the lane is same lane. In time slice 4, the distance has changed to be less than 20 metres, while the other criteria remain the same. In time slice 5, the distance is changed to less than 5 metres, due to the fact that the brake pedal is still ‘off’ and the speed is within the limit, the inference process will take place in time slice 6 *(t+1)* to predict the crash.

Modelling the above scenario using the proposed system could done by setting the states of the nodes according to the data provided by the sensors. Table 6.11 illustrates the combination of evidence in time slices 1-5. The system will predict the crash state at time slice 6 (i.e. future time).

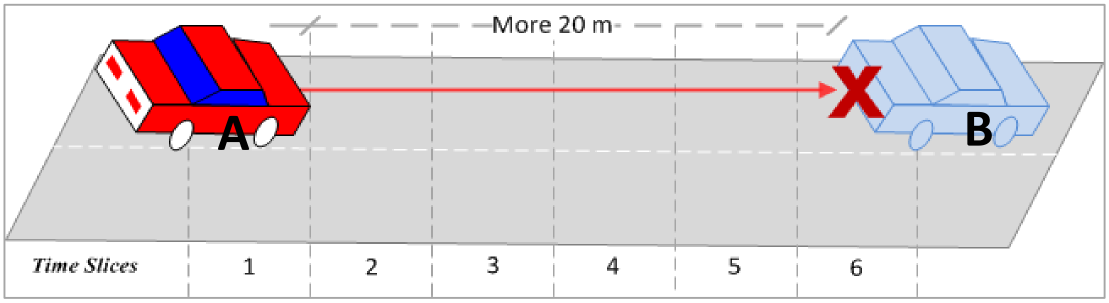

Figure 6.3 Vehicle is moving from point A to point B on a dual carriageway.

Time Slices		***1***	***2***	***3***	***4***	***5***	***6***
Scenario Inputs	Brake_ Pedal	OFF	OFF	OFF	OFF	OFF	To be predicted
	Distance	more20m	more20m	more20m	Less20m	Less5m	To be predicted
	Speed	Limit	limit	limit	Limit	limit	To be predicted
	Lane	Same	Same	Same	Same	Same	To be predicted
Crash belief and state of (t+1) time slice							
Fatal		-	-	-	-	-	-
Serious		-	-	-	-	**0.68**	**0.62**
Slight		-	-	-	-	-	-
No crash		**0.97**	**0.97**	**0.97**	**0.94**	-	-

Table 6.11 The second case theoretical scenario

Figure 6.3 demonstrates the inference results of the proposed DBN model after receiving the sensors' readings in time slices 1-5 in order to predict the crash at time slice 6. As shown in Table 6.11, the system has predicted no crash at time slice 1-3, with a belief of approximately 0.97. This is due to the fact that all the evidence from time slice 1 to 3 indicates that the distance is over 20 metres and that this is an indicator of no crash to the system, as the distance is more than the critical distance. At time slice 4, the distance is less than 20 metres, which is an indication to the DBN model to begin the inference process to establish the crash node degree of belief and predict the crash. The prediction is 0.94 no crash, due to the effect of the limit speed and the fact that the brake pedal was 'off'. At time slice 5, the prediction is 0.68 serious, as the distance is less than 5 metres, and the speed and brake pedal are still the same. Finally, the system predicts 0.62 serious crash at time slice 6 (t+1), as the system predicts that the distance will be less than 1 metre, the speed and brake pedal will still be in the same state of within the limit and ‘off’ respectively.

6.3.3 Experiment 3: Predicting the slight crash

The crash consider serious crash if the following criteria are meet, but not limited to them.

- The speed is below the limit.
- The distance is less than one metre.
- The lane is the same lane.
- The brake pedal is off.
- The likelihood of a slight crash may increase if the environment and driver are bad.

The following scenario will demonstrate the validity of the proposed system in predicting a slight crash. As shown in Figure 6.4, the vehicle is moving from point A to point B on a dual carriageway. The time during which the vehicle is moving between these points has been divided into 6 equivalent time slices, each representing a period of one second during which the vehicle collects new information via sensors and feeds it into the system. It can see from the figure that the vehicle and the one ahead are both moving in the same lane. From time slices, 1-3 the following criteria are present: the brake pedal is ‘off’, the distance is over 20 metres, the speed is below the limit and the lane is same. From time slice 4, the distance is change to less than 20 metres, the other criteria remaining the same. In time slice 5 the distance is changed to less than 5 metres due to the brake pedal still being ‘off’ and the speed below the limit. The inference process will take place in time slice 6 (future time) to predict the crash.

Modelling the above scenario using this proposed system could undertake by setting the states of the nodes according to the data provided by the sensors. Table 6.12 illustrates the combination of evidence in time slices 1-5. The system will predict the crash state at time slice 6.

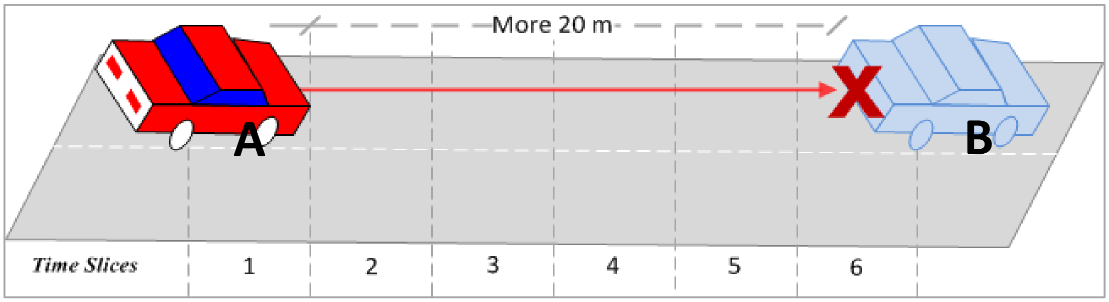

Figure 6.4 Vehicle is moving from point A to point B on a dual carriageway road

Time Slices		1	2	3	4	5	6
Scenario Inputs	Brake_ Pedal	OFF	OFF	OFF	OFF	OFF	To be predicted
	Distance	more20m	more20m	more20m	Less20m	Less5m	To be predicted
	Speed	Below_limit	Below_limit	Below_limit	Below_limit	Below_limit	To be predicted
	Lane	Same	Same	Same	Same	Same	To be predicted
Crash belief and state of (t+1) time slice							
Fatal		-	-	-	-	-	-
Siruous		-	-	-	-	-	-
Slight		-	-	-	-	0.78	0.69
No crash		0.97	0.97	0.97	0.98	-	-

Table 6.12 The first case theoretical scenario

Figure 6.4 demonstrates the inference results of the proposed DBN model after receiving the sensors reading in time slices 1-5 in order to predict the crash at time slice 6. It is could be seen from the figure, that there are four curves validating the degrees of belief in the crash node at different time slices. The system has predicted no crash at time slices 1-3 with a belief of approximately 0.97. This is due to the fact that all the evidence from time slices 1 to 3 indicates that the distance is over 20 metres and this is an indicator of no crash to the system as the distance is more than the critical distance.

At time slice 4, the distance is changed to be less than 20 metres, which is an indication to the to the DBN model to start the inferring process to find the crash node degree of belief and predict the crash. The prediction is 0.98 no crash due to the effect of being below the limit speed and the brake pedal being ‘off’. At time slice 5, the prediction is 0.78 as the distance is less than 5 metres, with the speed and brake pedal still in the same state. Finally, the system predicts a 0.69 slight crash at time slice 6 as the system predicts that the distance will be less than 1 metre, the speed and brake pedal will still in the same state of within the limit and ‘off’ respectively.

The prediction result must aid the driver or the vehicle to avoid the crash by appropriate action by the driver, or by applying avoiding application to prevent the crash.

6.4 System accuracy

In the above sections, we found the model results and analysed them. In this section, an accuracy evaluation is presented to check if the Pre-crash model accuracy depends partly on the 2011 Data set [1]. The 2011 data set contains 60,000 crash records; we used 50,000 records for training our DBN model, and then 10,000 records to evaluate our model. The GeNIe 2.0 application has the ability to validate the Bayesian network accuracy using real data sets. The evaluation results for the crash node in all time slices, (t-1, t, t+1), were as follows:

Crash level	Accuracy	Correct prediction	No. of unpredicted crashes	Total no. of DfT records
Fatal	0. 67899	624	295	919
Serious	0.55925	1,260	993	2,253
Slight	0.82285	2,513	541	3,054
No crash	0.98344	3,703	71	3,774
	Total	**8,100**	**1,900**	**10,000**
	Accuracy %	**81%**	**19%**	

Table 6. 13 The pre-crash system accuracy

In Table 6.13, the accuracy of the crash detection at time slice (t+1) is 81,000 / 10,000 = 0.81 and it is more accurate at time (t) and (t-1). Then, it starts to decrease at time t+1 because at this time slice not all data (events) are available, such as; the vehicle's speed and distance are not known at the future time (t+1) and this is normal in such systems which tried to predict the future state of the crash before it happens. The overall model accuracy is equal to 0.81 which is a good rate of accuracy.

6.5 Summary

This chapter has demonstrated the validity of the proposed system in predicting a crash and its severity in real time. The effect of the contributory factors of crash occurrences and their severity level had been shown clearly.

A number of experiments were introducing in order to evaluate the system’s ability to predict different levels of crash severity during driving. A detailed demonstration was given of the effect of the information and observable contexts in the system’s decision over time.

The prediction of vehicle crashes is very important to improving road safety and saving lives. The validation results of the proposed DBN model have revealed the ability of the system to predict crash occurrences of three levels of severity (fatal, serious and slight) accurately. In addition, the importance of including different sensor readings and taking into account the temporal aspect of the crash has been explain during this validation.

The next chapter (Chapter 7) presents the conclusion and proposed future related works.

Chapter 7 - Conclusion and Future Works

Chapter Objectives:

1. Summarise the work in this thesis.
2. Measure of success.
3. Propose future work to follow on from this thesis

7.1 Conclusions

Safety applications are attracting increasing attention as a promising area of VANET. Predicting crashes is vital to preventing accidents, which in turn saves lives. The main objective of this thesis is to establish a fundamental solution to preventing collisions from incidents and to create a safer environment by developing a pre-crash system in VANET, utilising a context-aware system approach.

The research into pre-crash systems in this thesis was presented in Chapter Two, and the crash prediction elements were defined in Chapter Three, based on a context-aware system approach. A five-layer context-aware pre-crash architecture for VANET was introduced in Chapter Four. This is able to predict the vehicle crash by sensing contextual information concerning the contributory factors (i.e driver, vehicle and environment). This consists of three phases: physical, thinking and application phases, which represent the three main subsystems of the context-aware layered conceptual framework: sensing, reasoning and acting subsystems. In the physical phase, the system senses information concerning all the contributory factors. The thinking phase is responsible for reasoning under uncertainty and over time, in order to predict the likelihood of a crash. Finally, the application phase is responsible for deploying the avoiding and/or warning applications.

The proposed pre-crash system deploys a dynamic Bayesian network as an uncertain reasoning algorithm in order to predict the likelihood of crashes at different levels of severity. Chapter Five presented a DBN model which was able to predict the likelihood of crashes at different levels of severity over time. Crash likelihood in the future is affected by the contributory factors at the current time and previous crash likelihood.

The proposed system was validated in Chapter Six using real high-fidelity data acquired from the DfT and historical values taken from different times and synthetic data. Different scenarios were used, with different initial values as an evaluation benchmark, in order to check the ways in which the DBN behaves, and to compare results against each other. These experiments demonstrate a high level of accuracy in detecting the likelihood and severity of crashes, particularly when compared with the analysis of the contributory data of the DfT. The findings of this study will help predict the probabilities of a crash at different severity levels, taking into account all factors relating to its the cause, which is an essential step towards creating a safer traffic system.

7.2 Measure of Success

This section focuses on each of the research questions outlined in Chapter One, in order to determine how successfully the proposed research solved these critical questions, as shown below:

The research questions specified in Chapter One were met as follows:

- ***How to design effective pre-crash system architecture for VANET by utilising a context-aware system approach.***

  A novel OBU pre-crash system architecture was designed in Chapter Four, based on the context-aware system approach to crash prediction in VANET. This architecture contains three main CAS phases: sensing, reasoning and application, depending directly on the sensed contextual information concerning the contributory factors.

- ***How to design an efficient pre-crash system that can carry out reasoning over time and under uncertainty?***

The objective of predicting crashes at different levels of severity was accomplished by establishing a novel DBN model in Chapter Five. The proposed DBN model combined contextual information concerning the contributory factors and performed reasoning over time and under uncertainty in order to effectively predict the likelihood of a crash.

- ***What kind of information is needed to accurately predict crash severity?***

The correct decision can be achieved by collecting increased contextual data using different kinds of sensors, taking into consideration all factors affecting everyday driving situations. The system proposed in this thesis uses real-time traffic and contextual information about the real contributory factors in order to predict crash likelihood and severity levels, as shown in Chapter Five.

- ***A study presenting the way in which this proposed architecture can be applied in VANET, in order to predict vehicle crashes, has to be conducted.***

Chapter Four presented the proposed OBU pre-crash system architecture and described the way in which these new components interact with each other, based on the main CAS components in order to predict the crash.

- ***An analysis of why DBN was chosen from among other reasoning techniques and a determination of the advantages of this technique must be performed.***

In Chapter Two, a comprehensive study was undertaken to look over available reasoning methods. The DBN method was selected from among these to be used in the proposed system. A fully detailed explanation of this reasoning method and its advantages and capabilities was provided in Chapter Three.

- ***A study showing the way in which the proposed system is different from others has to be demonstrated.***

In Chapter Two, a literature review concerning current pre-crash systems was conducted to present their disadvantages and limitations, in order to illustrate the differences between the current systems and the proposed system.

7.3 Future work

Nowadays, the road safety improvement and its efficiency in all the world; governments, vehicle manufacturers and researchers are seeking for new technologies, thereby preventing road crashes and thus reducing the number of fatalities and saving lives. Many challenges in this field of study still need to be covered in ITS.

In accordance to the proposed system carried out in this thesis, there are a number of concerns, which need to be taken into consideration in the future, as shown below:

- Extend the number of factors that affect the system decisions by adding more nodes (information and observable nodes) to the DBN in order to increase the prediction accuracy and system capability.

- Design an algorithm to be responsible for deploying the appropriate application according to the crash type and its severity.

- Design a mechanism to save the history of the driver behaviour, and implement DBN learning property.

- Extend the proposed system scope to predict the nearside and offside crashes, according to the DfT crash type classification.

- Design the dynamic context knowledge base to adjust and update the stored information with the driver behaviour and the historical crash data.

- Try to apply the proposed system in the real world to identifying each time slice in the DBN and to improve the time needed to sense the context, model it, and then process these contexts to deploy the prevention action.
- Design an algorithm to control the vehicle in the hazard situation either human controlling or auto control and what happens when both are reacting at the same time to avoid an accident.
- Expand the proposed system to be able reacting after predicting the crash, based on the crash severity level at first, and then deploy the suitable action to prevent or avoid the accident as a second step to be done (i.e. what is the stage after predicting a crash).
- Useful to share the results and the collected data with third party companies like TMC, Insurance companies and driver medical history to improve the decision makers.

Bibliography

[1] D. f. Transport. (2011). *Road casualties in Great Britain*. Available: https://www.esds.ac.uk/newRegistration/UKDSshowProjectDataset.asp?pn=63946

[2] D. f. Transport. (2010). *Road casualties in Great Britain*. Available: https://www.esds.ac.uk/newRegistration/UKDSshowProjectDataset.asp?pn=63947

[3] S. Olariu and M. C. Weigle, *Vehicular networks: from theory to practice* vol. 20: Chapman and Hall/CRC, 2010.

[4] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial intelligence*, vol. 29, pp. 241-288, 1986.

[5] A. Dey, G. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human–Computer Interaction*, vol. 16, pp. 97-166, 2001.

[6] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, pp. 380-392, 2014.

[7] S. Tokoro, K. Moriizumi, T. Kawasaki, T. Nagao, K. Abe, and K. Fujita, "Sensor fusion system for pre-crash safety system," *Proceedings in the IEEE on Intelligent Vehicles Symposium*, pp. 945-950, 2004.

[8] C. Campolo and A. Molinaro, "On vehicle-to-roadside communications in 802.11 p/WAVE VANETs," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1010-1015, 2011.

[9] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *Wireless Communications, IEEE*, vol. 13, pp. 36-43, 2006.

[10] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *Communications Surveys & Tutorials, IEEE*, vol. 11, pp. 3-20, 2009.

[11] Z. Zainol and K. Nakata, "Generic context ontology modelling: A review and framework," *Computer Technology and Development (ICCTD), 2010 2nd International Conference on*, pp. 126-130, 2010.

[12] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, pp. 263-277, 2007.

[13] Y. Jing, S. Yang, X. Wang, and W. Sun, "802.11-Based Multihop Ad Hoc Network Implementation," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, 2008, pp. 1-4.

[14] H. Moustafa and Y. Zhang, *Vehicular networks: techniques, standards, and applications*: Auerbach Publications, 2009.

[15] "Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager," *IEEE Std 1609.1*, pp. 1-71, 2006.

[16] W. Sheng-Wei and C. Meng-Yi, "Roadside units allocation algorithms for certificate update in VANET environments," in *Communications (APCC), 2011 17th Asia-Pacific Conference on*, 2011, pp. 472-477.

[17] S. Olariu and M. C. Weigle, *Vehicular networks: from theory to practice* vol. 20: Chapman & Hall/CRC, 2009.

[18] R. Baldessari, B. Bödekker, M. Deegener, A. Festag, W. Franz, C. Kellum, T. Kosch, A. Kovacs, M. Lenardi, and C. Menig, "CAR 2 CAR Communication Consortium Manifesto: Overview of the C2C-CC system, V1. 1," *Inst. Commun. Navig., Braunschweig, Germany, Tech. Rep. C2C-CC Manifesto,* 2007.

[19] "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services," *IEEE Std 1609.3-2007,* pp. 1-99, 2007.

[20] S. W. Wang and M. Y. Chang, "Roadside units allocation algorithms for certificate update in VANET environments," *Communications (APCC), 2011 17th Asia-Pacific Conference on,* pp. 472-477, 2011.

[21] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE,* vol. 46, pp. 164-171, 2008.

[22] A. Festag, A. Hessler, R. Baldessari, L. Le, W. Zhang, and D. Westhoff, "Vehicle-to-Vehicle and Road-Side sensor communication for enhanced road safety," in *Proceedings of the 15th World Congress on Intelligent Transport Systems*, 2008.

[23] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE,* vol. 2, pp. 12-22, 2007.

[24] V. Namboodiri, M. Agarwal, and L. Gao, "A study on the feasibility of mobile gateways for vehicular ad-hoc networks," *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks,* pp. 66-75, 2004.

[25] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," presented at the Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, Philadelphia, PA, USA, 2004.

[26] B. Fan and H. Krishnan, "Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE,* pp. 355-362, 17-20 Sept. 2006 2006.

[27] B. Fan and H. Krishnan, "Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications," *Intelligent Transportation Systems Conference, ITSC '06. IEEE,* pp. 355-362, 17-20 Sept. 2006.

[28] H. Menouar, F. Filali, and M. Lenardi, "A survey and qualitative analysis of mac protocols for vehicular ad hoc networks," *Wireless Communications, IEEE,* vol. 13, pp. 30-35, 2006.

[29] C. Cseh, "Architecture of the dedicated short-range communications (DSRC) protocol," in *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE*, 1998, pp. 2095-2099 vol.3.

[30] L. Garber, "Mobile WiMAX: The next wireless battle ground," *Computer,* vol. 41, pp. 16-18, 2008.

[31] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (VANETs): challenges and perspectives," *ITS Telecommunications Proceedings, 2006 6th International Conference on,* pp. 761-766, 2006.

[32] S. Kukliński and G. Wolny, "CARAVAN: Context-AwaRe Architecture for VANET," *Mobile Ad-Hock Networks: Applications*, pp. 221-244, 2011.

[33] A. Qureshi, J. Carlisle, and J. Guttag, "Tavarua: video streaming with wwan striping," *Proceedings of the 14th annual ACM international conference on Multimedia,* pp. 327-336, 2006.

[34] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto, "GVGrid: A QoS routing protocol for vehicular ad hoc networks," *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, pp. 130-139, 2006.

[35] C. Dobre and G. C. Tudor, "Mobile Advertisement in Vehicular Ad-Hoc Networks," *arXiv preprint arXiv:1202.2573,* 2012.

[36] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz, "CarTALK 2000: Safe and comfortable driving based upon inter-vehicle-communication," *Intelligent Vehicle Symposium, 2002. IEEE,* vol. 2, pp. 545-550, 2002.

[37] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: Applications and related technical issues," *Communications Surveys & Tutorials, IEEE,* vol. 10, pp. 74-88, 2008.

[38] I. Phase, "Cooperative intersection collision avoidance system limited to stop sign and traffic signal violations," 2008.

[39] B. Xu, S. Hischke, and B. Walke, "The role of ad hoc networking in future wireless communications," *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on,* vol. 2, pp. 1353-1358, 2003.

[40] G. M. T. Abdalla, M. A. Abu-Rgheff, and S. M. Senouci, "Current trends in vehicular ad hoc networks," *Ubiquitous Computing and Communication Journal,* 2007.

[41] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber, "A VANET-based emergency vehicle warning system," *Vehicular Networking Conference (VNC), 2009 IEEE,* pp. 1-8, 2009.

[42] H. Hartenstein and K. Laberteaux, *VANET: vehicular applications and inter-networking technologies* vol. 1: Wiley Online Library, 2010.

[43] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," *Proceeding of the IEEE on Vehicular Technology Conference (VTC),* pp. 2036-2040, 2008.

[44] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 9, pp. 69-72, 2005.

[45] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS),* vol. 10, pp. 91-102, 1992.

[46] D. Abowd, A. K. Dey, R. Orr, and J. Brotherton, "Context-awareness in wearable and ubiquitous computing," *Virtual Reality,* vol. 3, pp. 200-211, 1998.

[47] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network,* vol. 8, pp. 22-32, 1994.

[48] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach," *Proceedings of the 36th Annual Hawaii International Conference on System Sciences,* p. 10 pp., 6-9 Jan 2003.

[49] N. S. Ryan, J. Pascoe, and D. R. Morse, "Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant," *Computer Applications in Archaeology,* 1998.

[50] R. Hull, P. Neaves, and J. Bedford-Roberts, "Towards situated computing," *Wearable Computers. Digest of Papers., First International Symposium on,* pp. 146-153, 13-14 Oct. 1997.

[51] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," *Handheld and Ubiquitous Computing*, pp. 304-307, 1999.

[52] J. Indulska and P. Sutton, "Location management in pervasive systems," presented at the Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21, Adelaide, Australia, 2003.

[53] A. Schmidt and K. van Laerhoven, "How to build smart appliances?," *Personal Communications, IEEE*, vol. 8, pp. 66-71, 2001.

[54] T. Binh An, L. Young-Koo, and L. Sung-Young, "Modeling uncertainty in context-aware computing," *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*, pp. 676-681, 2005.

[55] H. Chen, T. Finin, and A. Joshi, "An intelligent broker for context-aware systems," *Adjunct Proceedings of Ubicomp*, vol. 3, pp. 183-184, 2003.

[56] T. Strang and C. Popien, "A Context Modeling Survey," in *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, 2004.

[57] B. Truong, Y.-K. Lee, and S.-Y. Lee, "Modeling and reasoning about uncertainty in context-aware systems," pp. 102-109, 2005.

[58] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, pp. 161-180, 2010.

[59] W. Dargie, "The Role of Probabilistic Schemes in Multisensor Context-Awareness," *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pp. 27-32, 19-23 March 2007 2007.

[60] Y. Vanrompay and Y. Berbers, "Opinion nets for reasoning with uncertain context information," *Contexts and Ontologies Representation and Reasoning*, p. 71, 2007.

[61] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments," *Pervasive Computing, IEEE*, vol. 3, pp. 62-70, 2004.

[62] A. K. Dey and G. D. Abowd, "CyberDesk: The use of perception in context-aware computing," *1st Workshop on Perceptual User Interfaces*, pp. 26-27, 1997.

[63] D. Salber, A. K. Dey, and G. D. Abowd, "The context toolkit: aiding the development of context-enabled applications," presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems, Pittsburgh, Pennsylvania, United States, 1999.

[64] C. M. Bishop, *Neural networks for pattern recognition*: Oxford university press, 1995.

[65] S. Bandini, A. Mosca, and M. Palmonari, "COMMON-SENSE SPATIAL REASONING FOR INFORMATION CORRELATION IN PERVASIVE COMPUTING," *Applied Artificial Intelligence*, vol. 21, pp. 405-425, 2007/04/25 2007.

[66] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor fusion using Dempster-Shafer theory [for context-aware HCI]," *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, vol. 1, pp. 7-12, 2002.

[67] Y. Zhang, Q. Ji, and C. G. Looney, "Active information fusion for decision making under uncertainty," in *Information Fusion, Proceedings of the Fifth International Conference on*, 2002, pp. 643-650.

[68] X. Li and Q. Ji, "Active affective state detection and user assistance with dynamic Bayesian networks," *Proceeding of the IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*, vol. 35, pp. 93-105, 2005.
[69] H. Wu, "Sensor data fusion for context-aware computing using dempster-shafer theory," Citeseer, 2003.
[70] Q. Ji, P. Lan, and C. Looney, "A probabilistic framework for modeling and real-time monitoring human fatigue," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 36, pp. 862-875, 2006.
[71] D. Salber, A. K. Dey, and G. D. Abowd, "Ubiquitous computing: Defining an hci research agenda for an emerging interaction paradigm," 1998.
[72] J. Pascoe, "Adding generic contextual capabilities to wearable computers," *Wearable Computers, Digest of Papers. Second International Symposium on*, pp. 92-99, 19-20 Oct. 1998.
[73] P. J. Brown, "Triggering information by context," *Personal Technologies*, vol. 2, pp. 18-27, 1998/03/01 1998.
[74] D. Salber, "Context-awareness and multimodality," *Colloque sur la multimodalité, IMAG, Grenoble*, 2000.
[75] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85-90, 1994.
[76] N. Davies, K. Mitchell, K. Cheverst, and G. Blair, "Developing a context sensitive tourist guide," *1st Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1*, 1998.
[77] S. Fickas, G. Kortuem, and Z. Segall, "Software organization for dynamic and adaptable wearable systems," *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, pp. 56-63, 1997.
[78] G. Kortuem, Z. Segall, and M. Bauer, "Context-aware, adaptive wearable computers as remote interfaces to'intelligent'environments," *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pp. 58-65, 1998.
[79] M. Brown, "Supporting user mobility," in *International Federation for Information Processing*, 1996.
[80] J. R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine, and W. A. S. Buxton, "Evolution of a reactive environment," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 170-177, 1995.
[81] S. Elrod, G. Hall, R. Costanza, M. Dixon, and J. Des Rivières, "Responsive office environments," *Communications of the ACM*, vol. 36, pp. 84-85, 1993.
[82] A. K. Dey, "Context-aware computing: The CyberDesk project," *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pp. 51-54, 1998.
[83] A. K. Dey, G. D. Abowd, and A. Wood, "CyberDesk: A framework for providing self-integrating context-aware services," *Knowledge-Based Systems*, vol. 11, pp. 3-13, 1998.
[84] P. J. Brown, J. D. Bovey, and C. Xian, "Context-aware applications: from the laboratory to the marketplace," *Personal Communications, IEEE*, vol. 4, pp. 58-64, 1997.
[85] P. J. Brown, "The Stick-e Document: a Framework for Creating Context-aware Applications," *Proceedings of EP'96, Palo Alto*, pp. 259-272, 1996.

[86] A. K. Dey, D. Salber, G. D. Abowd, and M. Futakawa, "The Conference Assistant: Combining Context-Awareness with Wearable Computing," presented at the Proceedings of the 3rd IEEE International Symposium on Wearable Computers, 1999.

[87] J. Pascoe, N. S. Ryan, and D. R. Morse, "Human computer giraffe interaction: HCI in the field," 1998.

[88] J. Rekimoto, Y. Ayatsuka, and K. Hayashi, "Augment-able Reality: Situated Communication through Physical and Digital Spaces," *ISWC '98: Proceedings of the 2nd IEEE International Symposium on Wearable Computers,* p. 68, 1998.

[89] S. Tokoro, K. Kuroda, A. Kawakubo, K. Fujita, and H. Fujinami, "Electronically scanned millimeter-wave radar for pre-crash safety and adaptive cruise control system," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 304-309, 2003.

[90] M. Skutek, M. Mekhaiel, and G. Wanielik, "A precrash system based on radar for automotive applications," *Proceedings of the IEEE Intelligent Vehicles Symposium,* pp. 37-41, 2003.

[91] M. Skutek, D. T. Linzmeier, N. Appenrodt, and G. Wanielik, "A precrash system based on sensor data fusion of laser scanner and short range radars," *Information Fusion, 8th International Conference on,* vol. 2, p. 8 pp., 2005.

[92] C. Kwanghyun, S. B. Choi, and L. Hyeongcheol, "Design of an Airbag Deployment Algorithm Based on Precrash Information," *Vehicular Technology, IEEE Transactions on,* vol. 60, pp. 1438-1452, 2011.

[93] N. F. M. Radzi, M. Mubin, and N. A. Rahim, "Collision detection algorithm for UAVs based on parametric theorem approach," *Communication Software and Networks (ICCSN), IEEE 3rd International Conference on,* pp. 493-496, 27-29 May 2011.

[94] P. Lindner and G. Wanielik, "Multi level fusion for an automotive pre-crash safety system," *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on,* pp. 143-146, 2008.

[95] H. D. Son, Y.-J. Kweon, and B. B. Park, "Development of crash prediction models with individual vehicular data," *Transportation Research Part C: Emerging Technologies,* vol. 19, pp. 1353-1363, 2011.

[96] R. Eigner and G. Lutz, "Collision Avoidance in VANETs - An Application for Ontological Context Models," *Pervasive Computing and Communications. PerCom 2008. Sixth Annual IEEE International Conference on*, pp. 412-416, 17-21 March 2008.

[97] R. Karlsson, J. Jansson, and F. Gustafsson, "Model-based statistical tracking and decision making for collision avoidance application," *American Control Conference,Proceedings of the 2004,* vol. 4, pp. 3435-3440 vol.4, June 30 2004.

[98] U. Lages, "Collision avoidance system for fixed obstacles-fuzzy controller network for robot driving of an autonomous vehicle," *Intelligent Transportation Systems, Proceedings. IEEE,* pp. 489-491, 2001.

[99] R. Moritz, "Pre-crash sensing-functional evolution based on short range radar sensor platform," in *SAE Conference*, 2000.

[100] R. K. Gnanamurthy, K. Saravanan, and A. Thangavelu, "Middleware architecture for intelligence in vehicular safety over VANET (InVANET)," *Proceeding of the International Conference on the Current Trends in Information Technology (CTIT),* pp. 1-7, 15-16 Dec 2009.

[101] M. Abdel-Aty and K. Haleem, "Analyzing angle crashes at unsignalized intersections using machine learning techniques," *Accident Analysis & Prevention,* vol. 43, pp. 461-470, 2011.

[102] F. Salim, S. Krishnaswamy, S. Loke, and A. Rakotonirainy, "Context-aware ubiquitous data mining based agent model for intersection safety," in *Proceeding of the EUC Workshops on Embedded and Ubiquitous Computing*, 2005, pp. 61-70.
[103] C. Kwanghyun, S. B. Choi, S. Kyungjae, and Y. Yousik, "A pre-crash discrimination system for an airbag deployment algorithm," *Preceeding in the American Control Conference (ACC)*, pp. 6949-6954, June 30 2010.
[104] C. Hsu, C. Liang, L. Ke, and F. Huang, "Verification of On-Line Vehicle Collision Avoidance Warning System using DSRC," *World Academy of Science, Engineering and Technology*, vol. 55, pp. 377-383, 2009.
[105] C. Caliendo, M. Guida, and A. Parisi, "A crash-prediction model for multilane roads," *Accident Analysis & Prevention*, vol. 39, pp. 657-670, 2007.
[106] Q. Tan, Q. Wei, J. Hu, and D. Aldred, "Road vehicle detection using fuzzy logic rule-based method," *Fuzzy Systems and Knowledge Discovery (FSKD), Seventh International Conference on*, vol. 3, pp. 1355-1358, 2010.
[107] (2013, 27 July). *stats19 Help File*. Available: http://www.stats19.org.uk/
[108] H. Lum and J. A. Reagan, "Interactive highway safety design model: Accident predictive module," *Public Roads*, vol. 58, pp. 14-17, 1995.
[109] S. Christmas and G. Britain, *The good, the bad and the talented: Young drivers' perspectives on good driving and learning to drive*: Department for Transport, 2007.
[110] K. Emmerson, *Learning to drive: the evidence*: Department for Transport, 2008.
[111] S. Al-Sultan, A. Al-Bayatti, and H. Zedan, "Context Aware Driver Behaviour Detection System in Intelligent Transportation Systems (ITS)," *Vehicular Technology, IEEE Transactions on*, vol. PP, pp. 1-1, 2013.
[112] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," University of California, 2002.
[113] T. Sando, "Modeling Highway Crashes Using Bayesian Belief Networks Technique and Gis," 2005.
[114] M. Jordan, "An introduction to probabilistic graphical models," *University of California, Berkeley*, 2003.
[115] T. L. Dean and K. Kanazawa, "Probabilistic Temporal Reasoning," in *AAAI*, 1988, pp. 524-529.
[116] G. Zheng, J. Wu, and L. Li, "Stochastic Process Algebra with Value-Passing," *Computer Science and Software Engineering, 2008 International Conference on*, vol. 3, pp. 374-377, 2008.
[117] G. Yang, Y. Lin, and P. Bhattacharya, "A driver fatigue recognition model based on information fusion and dynamic Bayesian network," *Information Sciences*, vol. 180, pp. 1942-1954, 2010.
[118] E. Charniak, "Bayesian networks without tears," *AI magazine*, vol. 12, p. 50, 1991.
[119] D. F. Preusser, A. F. Williams, S. A. Ferguson, R. G. Ulmer, and H. B. Weinstein, "Fatal crash risk for older drivers at intersections," *Accident Analysis & Prevention*, vol. 30, pp. 151-159, 1998.
[120] A. K. Pradhan, K. R. Hammel, R. DeRamus, A. Pollatsek, D. A. Noyce, and D. L. Fisher, "Using eye movements to evaluate effects of driver age on risk perception in a driving simulator," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 47, pp. 840-852, 2005.

[121] J. Carson and F. Mannering, "The effect of ice warning signs on ice-accident frequencies and severities," *Accident Analysis & Prevention*, vol. 33, pp. 99-109, 2001.
[122] J. B. Edwards, "The Relationship Between Road Accident Severity and Recorded Weather," *Journal of Safety Research*, vol. 29, pp. 249-262, 1998.
[123] C. f. A. R. R. S.-Q. C.-. Q), "Speeding: State of the Road," Queensland University of Technology, Australia2011.
[124] H. Renski, A. J. Khattak, and F. M. Council, "Effect of speed limit increases on crash injury severity: analysis of single-vehicle crashes on North Carolina interstate highways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1665, pp. 100-108, 1999.
[125] M. Hoedemaeker and K. A. Brookhuis, "Behavioural adaptation to driving with an adaptive cruise control (ACC)," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 1, pp. 95-106, 1998.
[126] D. N. Godbole, R. Sengupta, J. Misener, N. Kourjanskaia, and J. B. Michael, "Benefit evaluation of crash avoidance systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1621, pp. 1-9, 1998.
[127] M. A. Abdel-Aty and A. E. Radwan, "Modeling traffic accident occurrence and involvement," *Accident Analysis & Prevention*, vol. 32, pp. 633-642, Sep 2000.
[128] D. Heckerman, *A tutorial on learning with Bayesian networks*: Springer, 2008.
[129] D. Koller and N. Friedman, "Bayesian networks and beyond," *Unpublished manuscript*, 2004.
[130] K. Murphy, "The bayes net toolbox for matlab," *Computing science and statistics*, vol. 33, pp. 1024-1034, 2001.
[131] J. Bilmes and G. Zweig, "The Graphical Models Toolkit: An open source software system for speech and time-series processing," *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, vol. 4, pp. IV-3916-IV-3919, 2002.
[132] J. Bilmes and R. Rogers. (2014, 1 April). *The Graphical Models Toolkit (GMTK)*. Available: http://ssli.ee.washington.edu/~bilmes/gmtk
[133] L. Jouffe and P. Munteanu. (2014, 1 April). *Bayesia*. Available: http://www.bayesia.com/
[134] N. S. Corp. (1 April). *Norsys Software Corporation*. Available: http://www.norsys.com/
[135] D. S. Laboratory. (2014, 5 April). *GeNIe and SMILE*. Available: http://genie.sis.pitt.edu/wiki/GeNIe_Documentation
[136] S. Loke, *Context-aware pervasive systems: architectures for a new breed of applications*: CRC Press, 2006.
[137] E. Dilge, M. Gulbins, T. Ohnesorge, and B. Straube, "On a redundant diversified steering angle sensor," in *On-Line Testing Symposium, IOLTS. 9th IEEE*, 2003, pp. 191-196.
[138] G. L. Foresti and C. S. Regazzoni, "Multisensor data fusion for autonomous vehicle navigation in risky environments," *Proceeding of the IEEE Transactions on Vehicular Technology*, vol. 51, pp. 1165-1185, 2002.
[139] M. Bertozzi, A. Broggi, M. Felisa, G. Vezzoni, and M. Del Rose, "Low-level Pedestrian Detection by means of Visible and Far Infra-red Tetra-vision," in *Intelligent Vehicles Symposium, 2006 IEEE*, 2006, pp. 231-236.
[140] J. A. Misener, R. Sengupta, and H. Krishnan, "Cooperative collision warning: Enabling crash avoidance with wireless technology," *12th World Congress on ITS*, vol. 3, 2005.

[141] V. Stavroulaki, Y. Kritikou, and P. Demestichas, "Introducing cognition in the management of equipment in the future wireless world," in *Personal, Indoor and Mobile Radio Communications, PIMRC. IEEE 19th International Symposium on*, 2008, pp. 1-5.

[142] M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "Context aware architecture for sending adaptive HELLO messages in VANET," presented at the Proceedings of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems, Copenhagen, Denmark, 2010.

[143] N. Brahmi, L. Boukhatem, N. Boukhatem, M. Boussedjra, N. D. Nuy, H. Labiod, and J. Mouzna, "End-to-end routing through a hybrid ad hoc architecture for V2V and V2I communications," *Ad Hoc Networking Workshop (Med-Hoc-Net), The 9th IFIP Annual Mediterranean*, pp. 1-8, 2010.

[144] J. Pearl, "Belief networks revisited," *Artificial intelligence in perspective*, pp. 49-56, 1994.

[145] K. Livescu, O. Cetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, and A. Bezman, "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the JHU summer workshop," *Acoustics, Speech and Signal Processing, ICASSP. IEEE International Conference on*, vol. 4, pp. IV-621-IV-624, 2007.

[146] T. Åkerstedt and G. Kecklund, "Age, gender and early morning highway accidents," *Journal of Sleep Research*, vol. 10, pp. 105-110, Jun 2001.

[147] D. L. Massie, K. L. Campbell, and A. F. Williams, "Traffic accident involvement rates by driver age and gender," *Accident Analysis & Prevention*, vol. 27, pp. 73-87, 1995.

[148] C. Owsley, K. Ball, G. McGwin Jr, M. E. Sloane, D. L. Roenker, M. F. White, and E. T. Overley, "Visual processing impairment and risk of motor vehicle crash among older adults," *JAMA: the journal of the American Medical Association*, vol. 279, pp. 1083-1088, 1998.

[149] G. A. Davis, S. Davuluri, and J. Pei, "Speed as a risk factor in serious run-off-road crashes: Bayesian Case-Control Analysis with Case Speed Uncertainty," *Journal of Transportation and Statistics*, vol. 9, p. 17, 2006.

[150] C. KLOEDEN and J. MCLEAN, "SPEED AND THE RISK OF INVOLVEMENT IN A CASUALTY CRASH," *Road safety research, policing, education, conference*, pp. p. 188-90, 1998.

[151] D. F. Preusser, A. F. Williams, and R. G. Ulmer, "Analysis of fatal motorcycle crashes: Crash typing," *Accident Analysis & Prevention*, vol. 27, pp. 845-851, 1995.

[152] A. Niehaus and R. F. Stengel, "Rule-Based Guidance for Vehicle Highway Driving in the Presence of Uncertainty," *American Control Conference*, pp. 3119-3124, 26-28 June 1991 1991.

[153] R. P. Mahapatra, K. V. Kumar, G. Khurana, and R. Mahajan, "Ultra Sonic Sensor Based Blind Spot Accident Prevention System," *Proceeding of the ICACTE International Conference on Advanced Computer Theory and Engineering*, pp. 992-995, 20-22 Dec. 2008.

[154] D. P. Racine, N. B. Cramer, and M. H. Zadeh, "Active blind spot crash avoidance system: A haptic solution to blind spot collisions," *Proceeding of the IEEE International Symposium on Haptic Audio-Visual Environments and Games (HAVE)*, pp. 1-5, 16-17 Oct. 2010.

[155] W. Bing-Fei, H. Hao-Yu, C. Chao-Jung, C. Ying-Han, C. Chia-Wei, and C. Yen-Lin, "The blind spot warning system for daytime and nighttime," *Proceeding of the IEEE*

International Conference on Industrial Technology (ICIT), pp. 591-597, 19-21 March 2012.

[156] J. Saboune, M. Arezoomand, L. Martel, and R. Laganiere, "A Visual Blindspot Monitoring System for Safe Lane Changes," *Image Analysis and Processing – ICIAP*, vol. 6979, pp. 1-10, 2011.

Appendix A

This Appendix contains a full description of the DBN model nodes CPT's. As shown in chapter 5 the DBN nodes are classifies into three main kinds; information node, observable nodes and the hypothesis node. Therefore we will show the CPT's of all the proposed DBN nodes in three main classes.

A.1 Information Nodes – Page 85 – Group 1:

1- Age Node

AGE	
UNDER_18	0.30542
BTW_19_35	0.00200
BTW_36_65	0.08437
BTW_66_75	0.32527
OVER_75	0.28293

Table A. 1 Prior probability for AGE node

2- Gender Node

GENDER	
Male	0.20344
Female	0.79656

Table A. 2 Prior probability for GENDER node

3- Driver Node

DRIVER										
GENDER	Male					Female				
AGE	UNDER_18	BTW_19_35	BTW_36_65	BTW_66_75	OVER_75	UNDER_18	BTW_19_35	BTW_36_65	BTW_66_75	OVER_75
GOOD	0.30600	0.10350	0.37330	0.54211	0.63249	0.41494	0.69014	0.48954	0.84212	0.24164
BAD	0.69400	0.89650	0.62670	0.45789	0.36751	0.58506	0.30986	0.51046	0.15788	0.75836

Table A. 3 Prior probability for DRIVER node

4- Day of the week Node

DAY_OF_WEEK	
SUN	0.13231
MON	0.09840
TUE	0.14949
WED	0.29400
THU	0.16845
FRI	0.09891
SAT	0.05844

Table A. 4 Prior probability for DAY_OF_WEEK node

5- Time Node:

TIME	
PEAK_TIME	0.43058
OUT_PEAK_TIME	0.56942

Table A. 5 Prior probability for Time node

6- Traffic Status Node:

TRAFFIC_STATUS							
TIME	PEAK_TIME						
DAY_OF_WEEK	SUN	MON	TUE	WED	THU	FRI	SAT
DENSITY	0.37740	0.24321	0.34903	0.65110	0.81450	0.14000	0.52368
NORMAL	0.62260	0.75679	0.65097	0.34890	0.18550	0.86000	0.47632
TIME	OUT_PEAK_TIME						
DAY_OF_WEEK	SUN	MON	TUE	WED	THU	FRI	SAT
DENSITY	0.68616	0.87498	0.75834	0.56618	0.04278	0.48276	0.86960
NORMAL	0.31384	0.12502	0.24166	0.43382	0.95722	0.51724	0.13040

Table A. 6 Prior probability for TRAFFIC_STATUS node

7- Weather Conditions Node:

WEATHER_CONDITIONS	
Fine_AND_high_winds	0.19070
Fine_NO_high_winds	0.14938
Fog_or_mist	0.16456
Raining_AND_high_winds	0.27045
Raining_NO_high_winds	0.18127
Snowing_AND_high_winds	0.03132
Snowing_NO_high_winds	0.01232

Table A. 7 Prior probability for WEATHER_CONDITIONS node

8- Road Surface Conditions Node

ROAD_SURFACE_CONDITIONS	
Dry	0.10093
Flood_over_3cm_deep	0.29149
Frost_or_ice	0.18236
Snow	0.13334
Wet_or_damp	0.29188

Table A. 8 Prior probability for ROAD_SURFACE_CONDITIONS node

9- Light Condition Node:

LIGHT_CONDITIONS	
Daylight	0.58044
Darkness	0.41956

Table A. 9 Prior probability for WEATHER_CONDITIONS node

10 – Environment Node:

ENVIRONMENT																				
WEATHER_ CONDITIONS	Fine_AND_high_winds																			
LIGHT_ CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				
ROAD_SURFACE_ CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp
GOOD	0.57768	0.62346	0.16985	0.44350	0.57848	0.41258	0.38505	0.74091	0.38239	0.36067	0.34505	0.42844	0.22914	0.57229	0.15103	0.28628	0.33625	0.52203	0.09745	0.37421
BAD	0.42232	0.37654	0.83015	0.55650	0.42152	0.58742	0.61495	0.25909	0.61761	0.63933	0.65495	0.57156	0.77086	0.42771	0.84897	0.71372	0.66375	0.47797	0.90255	0.62579
WEATHER_ CONDITIONS	Fine_NO_high_winds																			
LIGHT_ CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				
ROAD_SURFACE_ CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp
GOOD	0.51963	0.44675	0.49385	0.46159	0.42498	0.30787	0.45416	0.37270	0.42364	0.29559	0.20526	0.11442	0.56007	0.41496	0.62342	0.64009	0.73366	0.56594	0.58265	0.35195

BAD	0.48037	0.55325	0.50615	0.53841	0.57502	0.69213	0.54584	0.62730	0.57636	0.70441	0.79474	0.88558	0.43993	0.58504	0.37658	0.35991	0.26634	0.43406	0.41735	0.64805
WEATHER_CONDITIONS	Fog_or_mist																			
LIGHT_CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				
ROAD_SURFACE_CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp
GOOD	0.46728	0.26148	0.33544	0.58168	0.13579	0.50111	0.62464	0.39032	0.45767	0.37456	0.65777	0.52889	0.36877	0.37517	0.37006	0.26233	0.60392	0.62707	0.36585	0.43307
BAD	0.53272	0.73852	0.66456	0.41832	0.86421	0.49889	0.37536	0.60968	0.54233	0.62544	0.34223	0.47111	0.63123	0.62483	0.62994	0.73767	0.39608	0.37293	0.63415	0.56693
WEATHER_CONDITIONS	Raining_AND_high_winds																			
LIGHT_CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				
ROAD_SURFACE_CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp
GOOD	0.56275	0.84507	0.40839	0.39227	0.29592	0.46681	0.44347	0.23771	0.42186	0.44931	0.49590	0.31315	0.35741	0.56304	0.54913	0.55062	0.12271	0.70835	0.54722	0.25760
BAD	0.43725	0.15493	0.59161	0.60773	0.70408	0.53319	0.55653	0.76229	0.57814	0.55069	0.50410	0.68685	0.64259	0.43696	0.45087	0.44938	0.87729	0.29165	0.45278	0.74240
WEATHER_CONDITIONS	Raining_NO_high_winds																			
LIGHT_CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				

ROAD_SURFACE_CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp
GOOD	0.21669	0.23495	0.79511	0.78545	0.25971	0.56757	0.32703	0.42517	0.34157	0.43555	0.59373	0.56693	0.61529	0.28736	0.08200	0.44226	0.78739	0.52011	0.48825	0.31568
BAD	0.78331	0.76505	0.20489	0.21455	0.74029	0.43243	0.67297	0.57483	0.65843	0.56445	0.40627	0.43307	0.38471	0.71264	0.91800	0.55774	0.21261	0.47989	0.51175	0.68432
WEATHER_CONDITIONS	Snowing_AND_high_winds																			
LIGHT_CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				
ROAD_SURFACE_CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp
GOOD	0.53277	0.35264	0.56627	0.22323	0.33803	0.38132	0.36403	0.80412	0.69623	0.35335	0.35019	0.13914	0.58951	0.32747	0.35420	0.53802	0.24847	0.71471	0.32971	0.35742
BAD	0.46723	0.64736	0.43373	0.77677	0.66197	0.61868	0.63597	0.19588	0.30377	0.64665	0.64981	0.86086	0.41049	0.67253	0.64580	0.46198	0.75153	0.28529	0.67029	0.64258
WEATHER_CONDITIONS	Snowing_NO_high_winds																			
LIGHT_CONDITIONS	Daylight										Darkness									
DAY_OF_WEEK	DENSITY					NORMAL					DENSITY					NORMAL				
ROAD_SURFACE_CONDITIONS	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp	Dry	Flood_over_3cm_deep	Frost_or_ice	Snow	Wet_or_damp

GOOD	0.17159	0.33993	0.54537	0.38407	0.49606	0.76631	0.41517	0.19870	0.23137	0.32843	0.36603	0.44894	0.24547	0.14828	0.65199	0.39928	0.28366	0.31013	0.74320	0.32252
BAD	0.82841	0.66007	0.45463	0.61593	0.50394	0.23369	0.58483	0.80130	0.76863	0.67157	0.63397	0.55106	0.75453	0.85172	0.34801	0.60072	0.71634	0.68987	0.25680	0.67748

Table A. 10 Prior probability for ENVIRONMENT node

11- Road Type Node:

ROAD_TYPE	
Dual_carriageway	0.4485701
One_way_street	0.0328321
One_way_street_Slip_road	0.0034072
Roundabout	0.0921666
Single_carriageway	0.3218616
Slip_road	0.1011631

Table A. 11 Prior probability for ROAD_TYE node

A.2 The Observable Nodes – Page 88 – Group 2:

1- Brake Pedal Node:

BRAKE_Pedal	
ON	0.37400
OFF	0.62600

Table A. 12 Prior probability for BRAKE_Pedal node

2- Distance Node at time slice (t):

DISTANCE								
CRASH	Fatal		Serious		Slight		NO_CRASH	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
less20m	0.00040	0.00040	0.01724	0.01905	0.00001	0.01334	0.55814	0.59806
less5m	0.15200	0.15200	0.37931	0.29525	0.66067	0.50667	0.30698	0.30993
less1m	0.84760	0.84760	0.60344	0.68571	0.33932	0.48000	0.13488	0.09201

Table A. 13 Prior probability for DISTANCE node

Distance Node at time slice (t):

DISTANCE (t)						
CRASH	Fatal					
BRAKE_Pedal	ON			OFF		
(Self) (t-1)	less20m	less5m	less1m	less20m	less5m	less1m
less20m	0.99760	0.01585	0.00000	0.01875	0.00007	0.00000
less5m	0.00160	0.49205	0.19934	0.66250	0.00000	0.00735
less1m	0.00080	0.49210	0.80066	0.31875	0.99993	0.99265
CRASH	Serious					
BRAKE_Pedal	ON			OFF		
(Self) (t-1)	less20m	less5m	less1m	less20m	less5m	less1m
less20m	0.85710	0.00000	0.00000	0.00000	0.05102	0.00000
less5m	0.14287	0.14832	0.01371	1.00000	0.08163	0.01316
less1m	0.00002	0.85168	0.98629	0.00000	0.86735	0.98684
CRASH	Slight					
BRAKE_Pedal	ON			OFF		
(Self) (t-1)	less20m	less5m	less1m	less20m	less5m	less1m
less20m	0.01593	0.04082	0.00000	0.12904	0.02186	0.00000
less5m	0.98039	0.44901	0.08163	0.87096	0.22405	0.01389
less1m	0.00368	0.51017	0.91837	0.00000	0.75410	0.98611

CRASH	NO_CRASH					
BRAKE_Pedal	ON			OFF		
(Self) (t-1)	less20m	less5m	less1m	less20m	less5m	less1m
less20m	0.60733	0.02913	0.00000	0.36310	0.00800	0.00000
less5m	0.38743	0.76214	0.02941	0.63095	0.49601	0.01852
less1m	0.00524	0.20874	0.97059	0.00595	0.49599	0.98148

Table A. 14 Prior probability for DISTANCE (t) node

3- Speed Node at time slice (t-1):

SPEED (t-1)								
CRASH	Fatal		Serious		Slight		NO_CRASH	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
OVER_LIMIT	0.82031	0.79688	0.29237	0.37972	0.05702	0.00329	0.00116	0.00060
LIMIT	0.07031	0.15104	0.69915	0.61557	0.44298	0.52961	0.09375	0.08756
BELOW_LIMIT	0.10156	0.04688	0.00424	0.00236	0.49561	0.46382	0.06134	0.19384
Zero	0.00781	0.00521	0.00424	0.00236	0.00439	0.00329	0.84375	0.71800

Table A. 15 Prior probability for SPEED node

Speed Node at time slice (t):

SPEED (t)								
CRASH	Fatal							
BRAKE_Pedal	ON				OFF			
(Self) (t-1)	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero
OVER_LIMIT	0.14465	0.00000	0.00000	0.00000	1.00000	0.62774	0.00000	0.00100
LIMIT	0.71398	0.01184	0.75625	0.00000	0.00000	0.37226	0.37321	0.00014
BELOW_LIMIT	0.14137	0.98763	0.06095	0.00000	0.00000	0.00000	0.62679	0.37035
Zero	0.00000	0.00053	0.18280	1.00000	0.00000	0.00000	0.00000	0.62851
CRASH	Serious							
BRAKE_Pedal	ON				OFF			
(Self) (t-1)	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero
OVER_LIMIT	0.16284	0.00000	0.00000	0.00000	1.00000	0.25191	0.00000	0.00000
LIMIT	0.83714	0.36106	0.00010	0.00000	0.00000	0.74809	0.50934	0.00000
BELOW_LIMIT	0.00002	0.61117	0.13376	0.00000	0.00000	0.00000	0.49066	0.38461

Zero	0.00000	0.02777	0.86614	1.00000	0.00000	0.00000	0.00000	0.61539
CRASH	Slight							
BRAKE_Pedal	ON				OFF			
(Self) (t-1)	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero
OVER_LIMIT	0.12133	0.00000	0.00000	0.00000	1.00000	0.31746	0.00000	0.00000
LIMIT	0.81808	0.28358	0.03124	0.00000	0.00000	0.68254	0.23333	0.00000
BELOW_LIMIT	0.06059	0.68657	0.24999	0.00000	0.00000	0.00000	0.76667	0.44017
Zero	0.00000	0.02985	0.71876	1.00000	0.00000	0.00000	0.00000	0.55983
CRASH	NO_CRASH							
BRAKE_Pedal	ON				OFF			
(Self) (t-1)	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero	OVER_LIMIT	LIMIT	BELOW_LIMIT	Zero
OVER_LIMIT	0.12664	0.00000	0.00000	0.00000	0.99990	0.13667	0.00000	0.00000
LIMIT	0.74932	0.23882	0.04464	0.00000	0.00001	0.86333	0.19644	0.00000
BELOW_LIMIT	0.12404	0.74626	0.20536	0.00000	0.00009	0.00000	0.80356	0.31599

Zero	0.00000	0.01492	0.75000	1.00000	0.00000	0.00000	0.00000	0.68401

Table A. 16 Prior probability for SPEED (t) node

4- Lane Node:

LANE				
CRASH	Fatal	Serious	Slight	NO_CRASH
Same_Lane	1.00000	0.99794	0.99846	0.00001
Different_Lane	0.00000	0.00206	0.00154	0.99999

Table A. 17 Prior probability for LANE node

5- Steering Angle Change Node:

STEERING_ANGLE_CHANGE		
LANE	Same_Lane	Different_Lane
YES	0.47120	0.55949
NO	0.52880	0.44051

Table A. 18 Prior probability for STEERING_angle_CHANGE node

6- Blind Spot Node:

Blind_Spot		
LANE	Same_Lane	Different_Lane
ON	0.47346	0.55150
OFF	0.52654	0.44850

Table A. 19 Prior probability for BLIND_SPOT node

A.3 The Hypothesis Node – Page 90

Crash Node at time slice (t-1)

CRASH (t-1)								
ROAD_TYPE	Dual_carriageway							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.14287	0.18513	0.12510	0.08016	0.19988	0.06905	0.08351	0.16670
Serious	0.00055	0.22210	0.12500	0.12001	0.05018	0.06903	0.12502	0.00020
Slight	0.21420	0.25918	0.37485	0.16018	0.29977	0.34471	0.33327	0.37487
NO_CRASH	0.64239	0.33359	0.37505	0.63966	0.45018	0.51721	0.45820	0.45822
ROAD_TYPE	One_way_street							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.00411	0.00263	0.10309	0.00644	0.00552	0.00981	0.45449	0.00361
Serious	0.00764	0.24489	0.15041	0.63440	0.32276	0.01824	0.01467	0.24528
Slight	0.32412	0.48743	0.15560	0.00973	0.31844	0.45501	0.01978	0.48712
NO_CRASH	0.66414	0.26505	0.59090	0.34943	0.35328	0.51694	0.51106	0.26398
ROAD_TYPE	One_way_street_Slip_road							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	

BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.10052	0.10052	0.00197	0.01775	0.01543	0.00386	0.00131	0.02100
Serious	0.01054	0.01054	0.01564	0.14080	0.12496	0.03124	0.32265	0.16243
Slight	0.18917	0.18917	0.46030	0.14274	0.02570	0.00643	0.32400	0.18400
NO_CRASH	0.69977	0.69977	0.52208	0.69872	0.83391	0.95848	0.35204	0.63257
ROAD_TYPE	Roundabout							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.00223	0.00109	0.00155	0.08329	0.18726	0.09096	0.00019	0.09513
Serious	0.16419	0.00045	0.19599	0.12489	0.12490	0.00018	0.07166	0.14282
Slight	0.00006	0.00003	0.19563	0.04167	0.06246	0.00009	0.07154	0.09524
NO_CRASH	0.83351	0.99842	0.60683	0.75015	0.62538	0.90876	0.85661	0.66681
ROAD_TYPE	Single_carriageway							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.02042	0.09999	0.07317	0.08046	0.11474	0.06202	0.06666	0.02609

Serious	0.14285	0.03334	0.24387	0.22988	0.21311	0.23256	0.23334	0.25217
Slight	0.08162	0.05555	0.02443	0.05748	0.16392	0.06977	0.13332	0.12174
NO_CRASH	0.75511	0.81111	0.65853	0.63219	0.50823	0.63566	0.56668	0.60000
ROAD_TYPE	Slip_road							
DRIVER	GOOD				BAD			
ENVIRONMENT	GOOD		BAD		GOOD		BAD	
BRAKE_Pedal	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Fatal	0.00148	0.00027	0.00837	0.00372	0.00226	0.00127	0.32208	0.16640
Serious	0.00131	0.00024	0.00049	0.00022	0.00025	0.00014	0.00508	0.32819
Slight	0.01246	0.32882	0.00055	0.19469	0.00443	0.14311	0.00129	0.00042
NO_CRASH	0.98475	0.67067	0.99058	0.80137	0.99307	0.85547	0.67155	0.50500

Table A. 20 Prior probability for CRASH node

Crash Node at time slice (t)

CRASH (t)																																
ROAD_TYPE	Dual_carriageway																															
DRIVE	GOOD																BAD															
ENVIRONMENT	GOOD								BAD								GOOD								BAD							
BRAKE_Pedal	ON				OFF				ON				OFF				ON				OFF				ON				OFF			
(Self) [t-1]	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH	Fatal	Serious	Slight	NO_CRASH
Fatal	0.00612	0.31375	0.00012	0.00000	0.99988	0.49630	0.00001	0.00000	0.20278	0.00028	0.00007	0.00000	0.99997	0.45243	0.00000	0.00000	0.01875	0.00222	0.00020	0.00000	0.85922	0.30714	0.05250	0.00000	0.47779	0.00028	0.00008	0.00000	0.99999	0.00234	0.00000	0.00000
Serious	0.66531	0.33125	0.08331	0.00000	0.00012	0.50370	0.27292	0.10000	0.60003	0.39723	0.18117	0.00000	0.00003	0.54757	0.59882	0.03579	0.97500	0.47778	0.00080	0.00000	0.14078	0.69286	0.42075	0.02863	0.51121	0.20278	0.09991	0.00000	0.00001	0.99766	0.34988	0.13038
Slight	0.32816	0.35000	0.25030	0.00000	0.00000	0.00000	0.72708	0.45011	0.19692	0.60003	0.09236	0.00000	0.00000	0.00000	0.40117	0.57137	0.00500	0.06667	0.11300	0.00000	0.00000	0.00000	0.52675	0.42863	0.00989	0.60003	0.20083	0.00000	0.00000	0.00000	0.65011	0.47833
NO_CRASH	0.00041	0.00500	0.66627	1.00000	0.00000	0.00000	0.00000	0.44989	0.00028	0.00247	0.72640	1.00000	0.00000	0.00000	0.00000	0.39284	0.00125	0.45333	0.88600	1.00000	0.00000	0.00000	0.00000	0.54275	0.00111	0.19692	0.69918	1.00000	0.00000	0.00000	0.00000	0.39128
ROAD_TYPE	One_way_street																															
DRIVE	GOOD																BAD															

ENVIRONMENT	BRAKE_Pedal	(Self) (t-1)	Fatal	Serious	Slight	NO_CRASH	ROAD_TYPE	DRIVE	ENVIRONMENT	BRAKE_Pedal	(Self) (t-1)
GOOD	ON	Fatal	0.30000	0.60000	0.08000	0.02000	One_way_street_Slip_road	GOOD	GOOD	ON	Fatal
		Serious	0.00500	0.07500	0.90000	0.02000					Serious
		Slight	0.00500	0.02000	0.82500	0.15000					Slight
		NO_CRASH	0.00000	0.00000	0.00000	1.00000					NO_CRASH
	OFF	Fatal	0.99000	0.01000	0.00000	0.00000				OFF	Fatal
		Serious	0.20000	0.80000	0.00000	0.00000					Serious
		Slight	0.00002	0.42656	0.57342	0.00000					Slight
		NO_CRASH	0.00000	0.00625	0.34375	0.65000					NO_CRASH
BAD	ON	Fatal	0.30010	0.60090	0.08900	0.01000			BAD	ON	Fatal
		Serious	0.00250	0.07502	0.90023	0.02225					Serious
		Slight	0.01000	0.08900	0.30010	0.60090					Slight
		NO_CRASH	0.00000	0.00000	0.00000	1.00000					NO_CRASH
	OFF	Fatal	0.99900	0.00100	0.00000	0.00000				OFF	Fatal
		Serious	0.01667	0.98333	0.00000	0.00000					Serious
		Slight	0.00025	0.07475	0.92500	0.00000					Slight
		NO_CRASH	0.00000	0.10000	0.52000	0.38000					NO_CRASH
GOOD	ON	Fatal	0.07500	0.90000	0.02000	0.00500		BAD	GOOD	ON	Fatal
		Serious	0.02000	0.30000	0.60000	0.08000					Serious
		Slight	0.00222	0.00889	0.47778	0.51111					Slight
		NO_CRASH	0.00000	0.00000	0.00000	1.00000					NO_CRASH
	OFF	Fatal	0.99000	0.01000	0.00000	0.00000				OFF	Fatal
		Serious	0.46667	0.53333	0.00000	0.00000					Serious
		Slight	0.44456	0.47778	0.07767	0.00000					Slight
		NO_CRASH	0.00000	0.01111	0.50000	0.48889					NO_CRASH
BAD	ON	Fatal	0.07502	0.90023	0.02225	0.00250			BAD	ON	Fatal
		Serious	0.00250	0.07502	0.90023	0.02225					Serious
		Slight	0.00250	0.02225	0.07502	0.90023					Slight
		NO_CRASH	0.00000	0.00000	0.00000	1.00000					NO_CRASH
	OFF	Fatal	0.99900	0.00100	0.00000	0.00000				OFF	Fatal
		Serious	0.78750	0.21250	0.00000	0.00000					Serious
		Slight	0.00004	0.01196	0.98800	0.00000					Slight
		NO_CRASH	0.00000	0.00625	0.34500	0.64875					NO_CRASH

Fatal	Serious	Slight	NO_CRASH	ROAD_TYPE	DRIVE	ENVIRONMENT	BRAKE_Pedal	(Self)(t-1)	Fatal	Serious	Slight	NO_CRASH
0.30000	0.60000	0.08000	0.02000	Roundabout	GOOD	GOOD	ON	Fatal	0.30000	0.60000	0.08000	0.02000
0.02000	0.30000	0.60000	0.08000					Serious	0.00500	0.07500	0.90000	0.02000
0.02000	0.08000	0.30000	0.60000					Slight	0.00500	0.02000	0.07500	0.90000
0.00000	0.00000	0.00000	1.00000					NO_CRASH	0.00000	0.00000	0.00000	1.00000
0.99000	0.01000	0.00000	0.00000				OFF	Fatal	0.99000	0.01000	0.00000	0.00000
0.20000	0.80000	0.00000	0.00000					Serious	0.20000	0.80000	0.00000	0.00000
0.00100	0.30000	0.69900	0.00000					Slight	0.00025	0.82500	0.17475	0.00000
0.00000	0.10000	0.50000	0.40000					NO_CRASH	0.00000	0.07704	0.38520	0.53776
0.30010	0.60090	0.08900	0.01000			BAD	ON	Fatal	0.82503	0.15023	0.02225	0.00250
0.01000	0.30010	0.60090	0.08900					Serious	0.00016	0.28594	0.71251	0.00139
0.01000	0.08900	0.30010	0.60090					Slight	0.00012	0.12456	0.37408	0.50125
0.00000	0.00000	0.00000	1.00000					NO_CRASH	0.00000	0.00000	0.00000	1.00000
0.99900	0.00100	0.00000	0.00000				OFF	Fatal	0.99994	0.00006	0.00000	0.00000
0.15000	0.85000	0.00000	0.00000					Serious	0.03750	0.96250	0.00000	0.00000
0.00006	0.01869	0.98125	0.00000					Slight	0.00003	0.39719	0.60278	0.00000
0.00000	0.02500	0.88000	0.09500					NO_CRASH	0.00000	0.12496	0.37523	0.49981
0.30000	0.60000	0.08000	0.02000		BAD	GOOD	ON	Fatal	0.25200	0.74400	0.00320	0.00080
0.02000	0.30000	0.60000	0.08000					Serious	0.00041	0.49592	0.50204	0.00163
0.02000	0.08000	0.30000	0.60000					Slight	0.24080	0.24320	0.25200	0.26400
0.00000	0.00000	0.00000	1.00000					NO_CRASH	0.00000	0.00000	0.00000	1.00000
0.99000	0.01000	0.00000	0.00000				OFF	Fatal	0.99960	0.00040	0.00000	0.00000
0.20000	0.80000	0.00000	0.00000					Serious	0.05000	0.95000	0.00000	0.00000
0.00100	0.30000	0.69900	0.00000					Slight	0.00002	0.16939	0.83059	0.00000
0.00000	0.10000	0.50000	0.40000					NO_CRASH	0.00000	0.07146	0.46433	0.46421
0.30010	0.60090	0.08900	0.01000			BAD	ON	Fatal	0.92223	0.06677	0.00989	0.00111
0.00250	0.82503	0.15023	0.02225					Serious	0.00250	0.07502	0.90023	0.02225
0.00250	0.02225	0.82503	0.15023					Slight	0.00016	0.00139	0.42656	0.57189
0.00000	0.00000	0.00000	1.00000					NO_CRASH	0.00000	0.00000	0.00000	1.00000
0.99900	0.00100	0.00000	0.00000				OFF	Fatal	0.99996	0.00004	0.00000	0.00000
0.03750	0.96250	0.00000	0.00000					Serious	0.28359	0.71641	0.00000	0.00000
0.00025	0.07475	0.92500	0.00000					Slight	0.00006	0.64369	0.35625	0.00000
0.00000	0.02500	0.88000	0.09500					NO_CRASH	0.00000	0.12498	0.40635	0.46867

ROAD_TYPE	DRIVE	ENVIRONMENT	BRAKE_Pedal	(Self) (t-1)	Fatal	Serious	Slight	NO_CRASH
Single_carriageway	GOOD	GOOD	ON	Fatal	0.20278	0.79444	0.00222	0.00056
				Serious	0.00008	0.26680	0.73281	0.00031
				Slight	0.00003	0.08333	0.37488	0.54176
				NO_CRASH	0.00000	0.00000	0.00000	1.00000
			OFF	Fatal	0.99995	0.00005	0.00000	0.00000
				Serious	0.00102	0.99898	0.00000	0.00000
				Slight	0.08681	0.30434	0.60885	0.00000
				NO_CRASH	0.00000	0.08065	0.43549	0.48387
		BAD	ON	Fatal	0.12716	0.87162	0.00110	0.00012
				Serious	0.00002	0.31815	0.68167	0.00017
				Slight	0.00004	0.21373	0.28578	0.50045
				NO_CRASH	0.00000	0.00000	0.00000	1.00000
			OFF	Fatal	1.00000	0.00000	0.00000	0.00000
				Serious	0.26655	0.73345	0.00000	0.00000
				Slight	0.04752	0.38078	0.57169	0.00000
				NO_CRASH	0.00000	0.02971	0.39605	0.57424
	BAD	GOOD	ON	Fatal	0.11300	0.77600	0.11080	0.00020
				Serious	0.00002	0.40615	0.56253	0.03129
				Slight	0.00002	0.02945	0.35290	0.61763
				NO_CRASH	0.00000	0.00000	0.01298	0.98702
			OFF	Fatal	0.99999	0.00001	0.00000	0.00000
				Serious	0.32603	0.67397	0.00000	0.00000
				Slight	0.00000	0.43737	0.56263	0.00000
				NO_CRASH	0.00000	0.09016	0.40984	0.49999
		BAD	ON	Fatal	0.00834	0.98891	0.00247	0.00028
				Serious	0.00001	0.28206	0.64100	0.07693
				Slight	0.00002	0.09523	0.23822	0.66653
				NO_CRASH	0.00000	0.00000	0.00000	1.00000
			OFF	Fatal	0.99999	0.00001	0.00000	0.00000
				Serious	0.24995	0.75005	0.00000	0.00000
				Slight	0.00000	0.28889	0.71111	0.00000
				NO_CRASH	0.00000	0.11111	0.42736	0.46153

ROAD_TYPE	DRIVE	ENVIRONMENT
Slip_road	GOOD	GOOD
		BAD
	BAD	GOOD
		BAD

BRAKE_Pedal	(Self) (t-1)	Fatal	Serious	Slight	NO_CRASH
ON	Fatal	0.30000	0.60000	0.08000	0.02000
	Serious	0.00500	0.82500	0.15000	0.02000
	Slight	0.00125	0.00500	0.64375	0.35000
	NO_CRASH	0.00000	0.00000	0.00000	1.00000
OFF	Fatal	0.99000	0.01000	0.00000	0.00000
	Serious	0.20000	0.80000	0.00000	0.00000
	Slight	0.00011	0.47778	0.52211	0.00000
	NO_CRASH	0.00000	0.28281	0.71094	0.00625
ON	Fatal	0.30010	0.60090	0.08900	0.01000
	Serious	0.01000	0.30010	0.60090	0.08900
	Slight	0.01000	0.08900	0.30010	0.60090
	NO_CRASH	0.00000	0.00000	0.00000	1.00000
OFF	Fatal	0.99900	0.00100	0.00000	0.00000
	Serious	0.15000	0.85000	0.00000	0.00000
	Slight	0.00004	0.25196	0.74800	0.00000
	NO_CRASH	0.00000	0.00123	0.37679	0.62198
ON	Fatal	0.30000	0.60000	0.08000	0.02000
	Serious	0.02000	0.30000	0.60000	0.08000
	Slight	0.00080	0.00320	0.73200	0.26400
	NO_CRASH	0.00000	0.00000	0.00000	1.00000
OFF	Fatal	0.99000	0.01000	0.00000	0.00000
	Serious	0.20000	0.80000	0.00000	0.00000
	Slight	0.00011	0.03333	0.96656	0.00000
	NO_CRASH	0.00000	0.00051	0.46173	0.53776
ON	Fatal	0.47779	0.51121	0.00989	0.00111
	Serious	0.00062	0.01876	0.97506	0.00556
	Slight	0.00250	0.02225	0.82503	0.15023
	NO_CRASH	0.00000	0.00000	0.00000	1.00000
OFF	Fatal	0.99975	0.00025	0.00000	0.00000
	Serious	0.15000	0.85000	0.00000	0.00000
	Slight	0.00006	0.33119	0.66875	0.00000
	NO_CRASH	0.00000	0.00204	0.66367	0.33429

Table A. 21 Prior probability for CRASH (t) node